NAVAL SHIP RESEARCH AND DEVELOPMENT CENTER
Bethesda, Maryland 20034

THE BANDIT COMPUTER PROGRAM FOR THE REDUCTION OF MATRIX BANDWIDTH FOR NASTRAN

Gordon C. Everstine

COMPUTATION AND MATHEMATICS DEPARTMENT
RESEARCH AND DEVELOPMENT REPORT

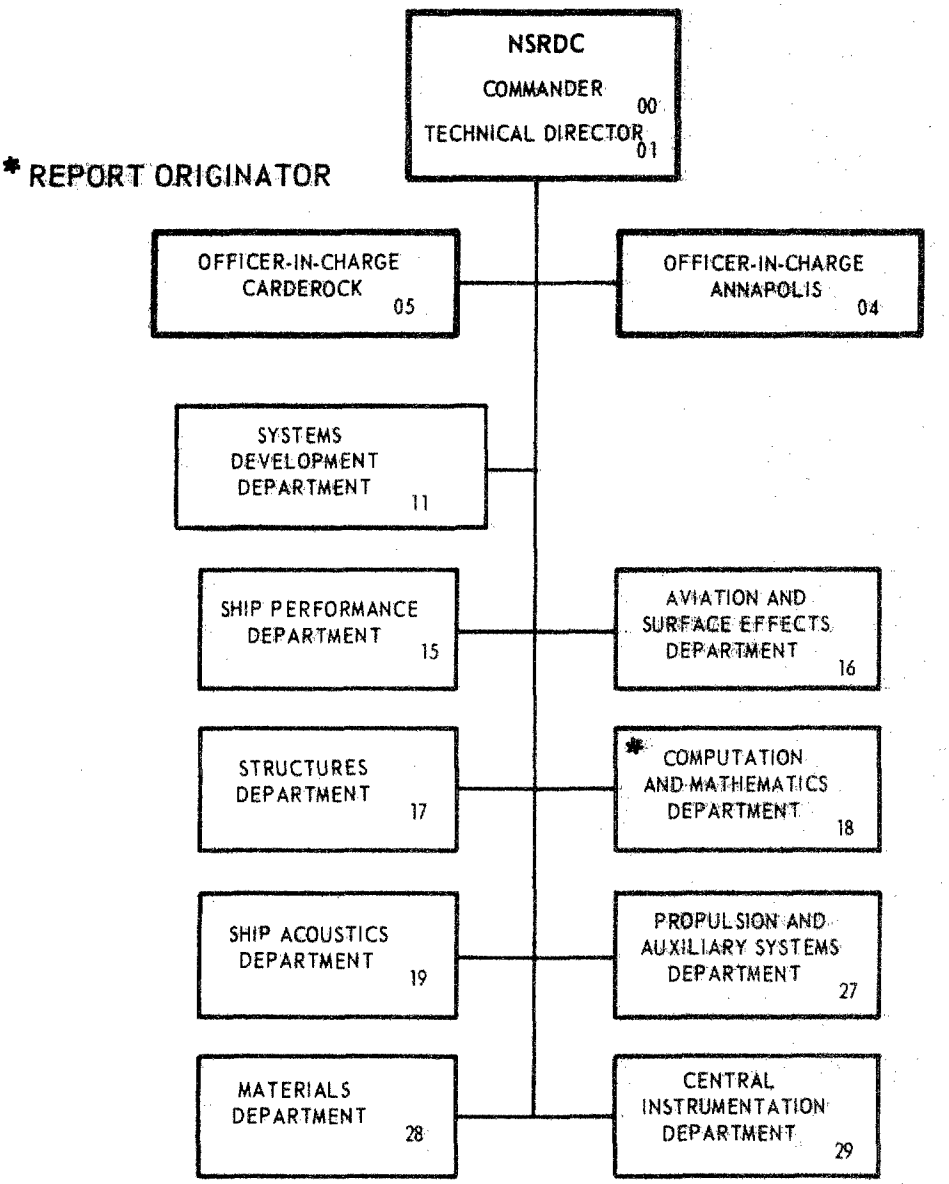
March 1972 Report 3827

THE BANDIT COMPUTER PROGRAM FOR THE REDUCTION OF MATRIX BANDWIDTH FOR NASTRAN

The Naval Ship Research and Development Center is a U. S. Navy center for laboratory effort directed at achieving improved sea and air vehicles. It was formed in March 1967 by merging the David Taylor Model Basin at Carderock, Maryland with the Marine Engineering Laboratory at Annapolis, Maryland.

Naval Ship Research and Development Center
Bethesda, Md. 20034

MAJOR NSRDC ORGANIZATIONAL COMPONENTS



DEPARTMENT OF THE NAVY
NAVAL SHIP RESEARCH AND DEVELOPMENT CENTER
Bethesda, Maryland 20034

THE BANDIT COMPUTER PROGRAM FOR THE REDUCTION OF MATRIX BANDWIDTH FOR NASTRAN

by

Gordon C. Everstine

March 1972

Report 3827

TABLE OF CONTENTS

LIST OF TABLES

ABSTRACT

This report describes a matrix bandwidth reduction preprocessor for use with the NASA structural analysis computer program, NASTRAN. Called BANDIT, the program is written in FORTRAN and uses the Cuthill-McKee strategy for resequencing grid points. Versions of the program for both CDC and other computers are presented.

ADMINISTRATIVE INFORMATION

The work reported herein was carried out under Task Area ZR 014 02 01.

I. INTRODUCTION

The NASA structural analysis computer program, NASTRAN, is a large general purpose program gaining wide acceptance in the Navy for the solution of both static and dynamic structural problems.

Since NASTRAN uses the finite element displacement method, the structural matrices which are formed are symmetric and sparse. With a suitable choice of the numbers (labels) assigned to the grid points, the matrices are also banded (i.e., the non-zero entries in each matrix are clustered about the main diagonal). For this reason, many of the routines used by NASTRAN for the solution of linear equations and for the extraction of eigenvalues operate most efficiently when the bandwidths of the structural matrices are minimum. Indeed, the number of calculations required in such routines is $O(n\,b^2)$, where n is the matrix order and b is the matrix bandwidth.

Although it is essential to the NASTRAN user to have matrices with small bandwidth, NASTRAN currently places the burden on the user to number his structure so as to provide such a bandwidth. The inherent difficulties in sequencing nodal labels manually and the increasing use of automatic data generators make this an excessive and unnecessary burden for most structural analysts.

NASTRAN currently allows the user to include in his input data deck a set of cards referred to as SEQGP cards. These cards define a look-up table giving the correspondence between the original grid numbers used in defining the problem and a new set of numbers to be used internally for all calculations.

This report describes a FORTRAN computer program called BANDIT which can be used as a preprocessor to the NASTRAN program to automatically resequence the grid point numbers for reduced bandwidth. Using a standard NASTRAN data deck as input, BANDIT resequences the numbering for reduced bandwidth, if possible, and generates a set of SEQGP punch cards for insertion into the NASTRAN deck.

The renumbering strategy used in BANDIT is that developed by Cuthill and McKee[1]. The need to resort to "strategies" becomes evident when one considers that n grid points (or nodes)[2] can be sequenced in n! distinct ways. Thus, with any strategy, there is no guarantee that an optimum numbering (i.e., one yielding minimum bandwidth) will be achieved. However, of several strategies tested[3] to date, the Cuthill-McKee approach[1] appears to be the most consistent for the reduction of matrix bandwidth for the classes of structures of prime interest to the Navy.

The computer program described herein was developed primarily for use on CDC 6400/6600 computers and hence has some machine-dependent features. However, for use on other computers, a machine-independent (and slightly less versatile) version of BANDIT is also described.

---

1 Cuthill, E. H. and J. M. McKee, "Reducing the Bandwidth of Sparse Symmetric Matrices," Proceedings of the 24th National Conference ACM 1969, pp. 157-172.

2 Throughout this report, "grid point" and "node" are used interchangeably.

3 "Sparse Matrices and Their Applications," Edited by D. J. Rose and R. A. Willoughby, Plenum Press, New York (1972), "Several Strategies for Reducing the Bandwidth of Matrices)," (E. H. Cuthill), pp. 157-166.

II. USE OF THE BANDIT PROGRAM

Throughout this and subsequent sections, it is assumed that the reader is familiar with the use of the NASTRAN structural analysis computer program[4].

BANDIT's primary reason for existence is the generation of the NASTRAN SEQGP data cards to effect low matrix bandwidth. As a by-product, BANDIT can also be used to right-adjust the NASTRAN bulk data. In either case, following the execution of BANDIT, the complete right-adjusted data deck is available on disk file. In addition, the user can elect to have punch card output for either the entire deck or the SEQGP cards alone.

The input data deck for BANDIT consists of a standard NASTRAN data deck (ID card through ENDDATA card, inclusive) with the addition of appropriate BANDIT option cards somewhere before the BEGIN BULK card. These cards, called $-option cards, indicate to BANDIT the user's choice of options, i.e., what the user wants BANDIT to do. The $-option cards are listed and described in detail in the next section.

On CDC machines, BANDIT functions as a variable-core program. Hence it is essentially open-ended with respect to the number of grid points that can be handled. During execution, the system is interrogated to determine the field length (amount of core). The dimensions of key arrays are then set so as to fill the available core. As a result, BANDIT must be executed on CDC machines with "NOREDUCE." in effect in order to prevent the automatic reduction of field length after the program is loaded.

---

4 "The NASTRAN User's Manual," edited by C.W. McCormick, NASA SP-222, September 1970.

BANDIT will load and execute in less than 50000_8[5] words of core. With this field length, typical structures with less than 500 grid points can be handled. For larger structures, more core may be needed, in which case BANDIT so informs the user. It has been our experience, however, that rarely are more than 60000_8 words needed. A more detailed discussion of core requirements appears in Section V.

Although BANDIT will accept an entire NASTRAN deck as input, resequencing requires only the following NASTRAN cards: BEGIN BULK, ENDDATA, and all "connection" cards. In particular, GRID cards are not used by BANDIT. The current list of connection cards which BANDIT recognizes is given in Table 1.

BANDIT will accept data on either short or long field data cards. The only restriction to the data concerns sorting. Since BANDIT does not sort the bulk data deck, each continuation to a connection card must immediately follow the parent card. Normally, however, unless long field cards are being used, each logical connection card consists of only one physical card.

If the user so indicates, BANDIT will process all multi-point constraint (MPC) cards present. While NASTRAN MPC's refer to individual degrees of freedom, BANDIT considers only grid points. Thus, each dependent point appearing in an MPC relation is eliminated from the connection table. Additional connections are also generated between each independent point in the constraint equation and every other point to which the dependent point was previously connected.

---

5 The subscript "8" in this context means "base 8".

TABLE 1 - CONNECTION CARDS RECOGNIZED BY BANDIT

CBAR	CIS2D4	CQUAD2
CCONEAX	CIS2D8	CQUAD3
CDAMP1	CIS3D8	CROD
CDAMP2	CIS3D20	CSHEAR
CDAMP3	CISH8	CTETRA
CDAMP4	CISH16	CTORDRG
CELAS1	CMASS1	CTRAPRG
CELAS2	CMASS2	CTRBSC
CELAS3	CMASS3	CTRIA1
CELAS4	CMASS4	CTRIA2
CFLUID2	CONM1	CTRIARG
CFLUID3	CONM2	CTRMEM
CFLUID4	CONROD	CTRPLT
CHEXA1	CQDMEM	CTUBE
CHEXA2	CQDPLT	CTWIST
CHTTRI2	CQUAD1	CVISC

It should be emphasized that only in rare cases would it make sense to let BANDIT process MPC's. The main reasons for this are that BANDIT does not consider individual degrees of freedom and, in addition, cannot distinguish one MPC "set" from another. Moreover, the effects of MPC's might be better handled by NASTRAN's active column feature.

The whole question of NASTRAN active columns complicates the bandwidth reduction problem, since there are clearly cases in which certain grid points should be relegated to active columns. An example might be the grid points common to both the fuselage and a wing on an airplane. If the user is able to identify such points, he can indicate them to BANDIT using the $IGNORE card described in the next section. This BANDIT feature, like the MPC feature, will probably find only occasional use.

Following the successful completion of a BANDIT run, whether resequencing was performed or not, the entire NASTRAN deck is contained on a file called TAPE8 (logical unit 8 on some machines). If resequencing has been performed, this file includes the SEQGP cards generated. These cards are inserted into the bulk data deck before the first card whose mnemonic would alphabetically follow "SEQGP". Thus, for a NASTRAN deck already properly sorted, the block of SEQGP cards will be inserted into its proper place. On machines such as the IBM 360, whose collating sequence is opposite to that on the CDC 6400, modification of the coding is needed for proper placement.

III. THE $-OPTION CARDS

The input data deck for BANDIT consists of a standard NASTRAN data deck (ID card through ENDDATA card, inclusive) with the addition of appropriate BANDIT option cards somewhere before the BEGIN BULK card. These option cards take the form of NASTRAN comment cards, i.e., a card with a dollar sign ($) in card column #1.

The BANDIT $-option cards may appear in any order and any location as long as they precede the BEGIN BULK card. There are two general formats for these cards,

$KEYWORD1 KEYWORD2

or

$KEYWORD1 N1 N2 N3 ... ,

where the Ni's are positive integers separated by one or more blanks. In order to qualify as a NASTRAN comment card, the $ must appear in card column #1.

Additional restrictions on the $-option cards are as follows:

(1) KEYWORD1 must start in card column #2.

(2) There may be no imbedded blanks in either keyword.

(3) Keywords (or integers) must be separated by one or more blanks.

(4) At least the first two letters of each keyword are required for proper identification.

A complete list of the $-option cards, along with a summary of the use of each card, appears in Table 2. In the absence of any card listed, the underlined option is chosen by default.

TABLE 2 - SUMMARY OF BANDIT $-OPTION CARDS

(Underline denotes default; only the first two letters of each keyword are required)

$SEQUENCE	NO	Resequencing not desired
	YES	Resequencing is desired
$RIGHTADJUST	NO	No right-adjusting desired
	YES	Bulk data is to be right-adjusted
$PUNCH	NONE	No punch output desired
	SEQGP	Only SEQGP cards are to be punched
	ALL	The entire NASTRAN deck is to be punched
$GRID	N1	The integer N1 is an upper bound on the number of grid points. (The default limits the maximum nodal degree to approximately 19.)
$PRINT	MIN	Basic printed output
	MAX	Extensive printed output
$MPC	NO	MPC cards are not to be processed
	YES	MPC cards are to be processed
$IGNORE	N1 N2 N3 ...	Grid numbers Ni appearing here are ignored during resequencing

The $SEQUENCE card is required to resequence the grid point labels and generate the SEQGP cards. For resequencing purposes, the only other data cards required are BEGIN BULK, ENDDATA, and all connection cards.

The right-adjusting of the bulk data is performed automatically if resequencing ($SEQUENCE YES) is elected. The user can then elect to have this deck punched by using $PUNCH ALL. In any case, it can be accessed on TAPE8.

The standard printed output consists of a title page, a listing of the SEQGP cards generated (if resequencing is requested), and a user summary. Use of the $PRINT MAX card results in the printing of additional tables as well as information on the flow of calculations during the actual resequencing. However, because of the additional work involved in generating several of the tables, the user pays a penalty in the form of increased execution time. A detailed explanation of BANDIT output appears in the next section.

Use of the $MPC YES card results in the processing of all MPC cards in the NASTRAN deck, regardless of their identifying set numbers. During processing, all dependent grid points on all MPC cards will be eliminated from the connection table after the additional connections due to the constraint relations are accounted for. For this reason, the user would normally decline to use this feature.

Although BANDIT is a variable-core program, the specific way in which the available core is partitioned depends on both the number of grid points and the maximum nodal degree. Nodal degree is defined in Section IV. B. Based on the space available, BANDIT computes default values for the dimensions of various arrays. This partitioning can be

optimized for larger problems by declaring to BANDIT the number of grid points present. The appropriate $-option card is $GRID N1. Here, N1 is an upper bound (preferably least upper bound) on the number of grid points. In the absence of this card, the default values computed by BANDIT result in a limit of approximately 19 on the maximum nodal degree.

The $IGNORE card (Table 2) can be used to designate those grid points Ni which should be ignored completely by BANDIT during resequencing. This normally results (in NASTRAN) in those points being placed into active columns. Any number of $IGNORE cards may appear, although the total number of ignored points may not exceed 100. Ignored points are renumbered last by the SEQGP cards.

IV. PRINTED OUTPUT

A. DESCRIPTION

There are two levels of BANDIT printed output: maximum printing (obtained by using the $PRINT MAX card), or minimum printing (obtained by default or by using $PRINT MIN). The latter is a subset of the former.

If resequencing is elected, the basic (minimum) output consists of a listing of the SEQGP cards generated and a user summary. The user summary contains the following information:

(1) original matrix semi-bandwidth

(2) new matrix semi-bandwidth

(3) central processor (CP) time in BANDIT, in seconds

(4) original matrix profile

(5) new matrix profile

(6) number of grid points

(7) number of elements

(8) number of components

(9) maximum nodal degree

(10) number of points of zero degree

(11) punch output requested

(12) field length (FL), octal

(13) the FORTRAN variables MAXGRD, MAXDEG, & KORE (defined below)

(14) date and time

In the "machine-independent" version of BANDIT (Appendix B), items 3 and 14 and KORE are omitted, since the determination of these quantities involves machine-dependent coding.

B. DEFINITIONS

For a matrix A, we follow the notation of Cuthill[6] and define θ_i as the difference between i and the column index of the first non-zero element of row i of A. Then the semi-bandwidth B is given by

$$B = \max_i \theta_i \tag{1}$$

This value is listed in 1 and 2 above. We note that the relationship between B and the "NASTRAN bandwidth" B_N is

$$B_N = (B+1)\,k\,, \tag{2}$$

6 Cuthill, E.C., op. cit.

where k is the average number of degrees-of-freedom per grid point. This formula assumes zero NASTRAN active columns.

The profile P of the matrix A is defined as

$$P = \sum_{i=1}^{N} \theta_i \tag{3}$$

where N is the matrix order. These values are listed in 4 and 5 of the user summary. They provide some measure of the space which would be required to store the matrix A if profile storage were employed instead of band storage. Since NASTRAN does make use of active columns in its routines, the matrix profile may be of interest to some users.

The number of grid points counted by BANDIT (and listed in the user summary) includes only those points appearing on recognizable elements (Table 1). The NASTRAN GRID cards are not processed.

The number of components of a structure is the number of independent substructures, each of which has no connections with grid points of any other substructure. In the event MPC's are processed, each dependent point is eliminated from the connection table and hence becomes its own component.

The degree of a grid point (node) is defined as the number of other grid points to which it is connected. The user summary lists both the maximum nodal degree and the number of grid points of zero degree.

The variables MAXGRD and MAXDEG are the upper bounds on the number of grid points and maximum nodal degree, respectively, for a given BANDIT run. The variable KORE (given in both octal and decimal) refers to the length of blank common in words. It is included in the summary to aid the user in determining his core requirements for very large structures. (See Section V.)

If the user elects maximum printing, the printed output also includes an internal/external grid point correspondence table, three connection tables, and a set of informational messages concerning the renumbering strategy.

The correspondence table lists, for each internal number, the original grid number to which it corresponds. These internal numbers are simply the integers 1 to N for a structure containing N grid points.

The three connection tables supply connectivity information in terms of internal numbers, original grid point numbers, and renumbered numbers (new numbers assigned by the SEQGP cards). For each node label i, the connection table lists its component index, the "distance" from the first non-zero entry in row i (of the matrix) to the diagonal (θ_i), the degree of node i, and the labels of the adjacent nodes.

V. CORE REQUIREMENTS ON THE CDC COMPUTERS

For a given structure, the core requirements depend on two parameters: the number of grid points NN, and the maximum nodal degree MM.

It is not intended that the user should normally have to calculate the required core in order to use BANDIT. A field length of about 55000_8 words should be sufficient for most structures. However, to cover situations in which either NN or MM is unusually large, the user can estimate his core requirements using the algorithm briefly described here.

The length of blank common storage during any given run is denoted KORE. This space is partitioned among several arrays whose dimensions are given in terms of the variables MAXGRD and MAXDEG, which are upper bounds on NN and MM, respectively. The approximate relationship between these variables is

$$\text{KORE} = (\text{MAXGRD}/\text{K}+3)*\text{MAXDEG}+ 8*\text{MAXGRD} , \qquad (4)$$

where K, the number of integers packed per word, is given by

$$\text{K} = \begin{cases} 6 & , \text{MAXGRD} < 510 \\ 4 & , \text{MAXGRD} > 2045 \\ 5 & , \text{otherwise.} \end{cases} \qquad (5)$$

In the absence of a $GRID card in the NASTRAN deck, BANDIT assigns default values to MAXGRD and MAXDEG such that MAXDEG is approximately 19. Thus, whenever the user anticipates a maximum nodal degree MM greater than 19, he must make use of the $GRID card.

Using Equation (4), structures for which KORE $< 8500_{10}$ can be run in a field length (FL) of 55000_8 words. Thus, for larger values of KORE, the user need only increase the FL accordingly.

VI. THE RENUMBERING STRATEGY

Although most of the FORTRAN coding in BANDIT is devoted to the task of developing the connection table, the heart of the program is the strategy used for renumbering. BANDIT uses the bandwidth-reduction

approach developed by E. H. Cuthill and J. M. McKee[7]. The resequencing subroutines were written during their early research in this area. Here, for completeness, we present a brief summary of the main ideas of the strategy. A complete discussion, including comparison with other methods, appears in the Cuthill-McKee paper.[7] A recent survey article by Cuthill[8] compares algorithms developed for reducing matrix bandwidth, wavefront, or profile. Extensive bibliographies appear in both these papers and hence need not be cited here.

For the purposes of this discussion, a starting node (or grid point) is one given the new label 1. The Cuthill-McKee method[7] is direct rather than iterative. It involves first the selection of one or more possible starting nodes. Although these nodes are normally of low degree, the one eventually chosen to be the starting node need not be of minimum degree.

For each possible starting node, the remaining nodes are relabeled according to the following prescription: The nodes adjacent to the starting node are labeled in sequence in the order of their increasing degree. In the terminology of graph theory, these nodes are said to be at the first level. Next, for each node of level 1 and in sequence, the numbering continues with those nodes as yet unnumbered and adjacent, in the order of their increasing degree. The set of all nodes (other than 1) adjacent to level 1 nodes thus constitute level 2. The numbering continues in this fashion, level-by-level, until all nodes have been numbered. If several nodes could receive a given label, the first node to qualify is chosen.

---

7 Cuthill, E. H. and J. M. McKee, op. cit.

8 Cuthill, E. H., op. cit.

This procedure is carried out for each possible starting node previously selected. The sequence yielding the lowest bandwidth is finally chosen.

It is apparent that, in the absence of ties for a given label, the relabeling sequence is independent of the original numbering once a starting node has been selected. Thus the original nodal numbering has almost no effect on the final numbering.

A secondary criterion used by BANDIT in the renumbering is the matrix profile P. (The definitions of P and the semi-bandwidth B were given in Section IV.) The BANDIT criterion is that, of those nodal numberings which yield the lowest B, the one resulting in the lowest P is chosen. This often has a beneficial effect because NASTRAN uses active columns in matrix factoring.

Accordingly, all nodes of zero degree are numbered last. A node of zero degree occurs in BANDIT either when selected directly by the user (on $IGNORE cards) or from MPC equations, in which case the dependent nodes are "eliminated" and thus given zero degree.

A final attempt at reducing the profile still further is made by reversing the previous best numbering; i.e., the nodes labeled 1, 2,..., n are relabeled n, n-1, ...,1. As pointed out by George[9], this frequently results in a lower profile P.

---

9 George, J.A., "Computer Implementation of the Finite Element Method," Ph.D. Thesis, Computer Science Department, Stanford University, 1971.

VII. DESCRIPTION OF THE CODING

The CDC 6400/6600 version of BANDIT consists of a main program, 20 FORTRAN subroutines, six FORTRAN functions, and two routines written in the COMPASS assembly language. The complete program listing is given in Appendix A.

A second version of BANDIT, listed in Appendix B, is essentially the same as the CDC version except that all aspects of the program which are unique to the CDC machines have been deleted. For example, the COMPASS routines, upon which the integer packing and variable-core features depend, and all timing routines have been removed. Thus, this version is slightly less general than the CDC version. On the other hand, the integer packing is less necessary on IBM and Univac machines whose word length is shorter than the 60-bit word length on CDC.

In this brief description of the coding for CDC machines, the differences between the two BANDIT versions are indicated, where appropriate.

The main program, called BANDIT in the CDC version, handles preliminary chores and controls some of the output printing and punching. CORSIZ is called to determine the amount of core available for blank common. GOOGAN is called to learn the user's choice of $-options and, if necessary, to right-adjust the NASTRAN bulk data deck. The partitioning of blank common is accomplished by a call from GOOGAN to GRID. BANDIT then calls NASNUM, which controls the complete processing of the NASTRAN deck as directed by the $-options chosen by the user. BANDIT's final duty is to control the listing of the NASTRAN deck, the punching of cards, and the printing of the user summary. BANDIT output was described in Section IV.

Subroutine NASNUM is the executive for the formal processing of the NASTRAN deck and is executed in its entirety only if resequencing is requested. NASNUM first reads the deck and forms the connection table IG. A temporary set of node numbers (denoted Set B) is first assigned in the order in which grid points are encountered. The user's original grid numbers constitute Set A. After all cards have been read, a permanent set of internal numbers (Set C) is assigned such that the user's original grid point numbers are arranged in ascending numerical order. If elected by the user, the connection table IG is then updated by subroutine TIGER to reflect the presence of MPC equations. Here, the new connections caused by MPC's are generated and the list of dependent grid points is saved. Then, with subroutine MORRIS, all dependent nodes and others chosen by the user on $IGNORE cards are deleted from IG. With the connection table IG now complete, the actual renumbering is performed by SCHEME, which generates a correspondence table between the Set C numbers and a new set of nodal numbers, Set D. The correspondence between the user's original numbers (Set A) and the new nodal numbers (Set D) appears on the SEQGP cards listed.

Subroutine FLIP converts an array of original grid point numbers to the internal numbers used by BANDIT. Only unique non-zero integers are retained in the list.

Subroutine GOOGAN reads a NASTRAN data deck (ID card through ENDDATA card, inclusive) and right-adjusts all bulk data. It optionally converts all cards with 8-column field widths to 16-column widths. It is used here to filter a data deck so as to retain only those cards associated with the structure geometry. Finally, it reads the user's $-option cards and sets the appropriate parameters.

Subroutine GRID sets up the dimensions of all those arrays whose lengths depend on either the number of grid points or the maximum nodal degree. The upper bounds on these two quantities are stored in MAXGRD and MAXDEG, respectively. Since the entries in the connection table IG are packed with either four, five, or six integers per word, the packing density is also determined here.

Subroutine READIT is called by GOOGAN whenever a $ card containing integer data is encountered. This routine interprets and stores in IP all positive integers on the data card. The variable NIP contains the number of integers stored in IP. A(I) contains the alphabetic representation of the character in card column I.

Subroutine BOMBIT is called whenever a fatal error is detected. This routine writes an appropriate error message onto both the output file, if necessary, and the CDC dayfile. The job is then aborted in order to suppress the execution of NASTRAN following that of BANDIT.

Subroutine SCAT is called once for each grid point appearing on a connection card. The routine supplies a fast way of determining for a grid point whether that point has been encountered before and, if so, which temporary internal number has been assigned to it. This is accomplished with the array INV(I, J), where INV(I, 1) contains an original grid number and INV(I, 2) contains the internal number assigned to it. The location I chosen for grid point N is given by

$$I = \text{MOD}(N-1, \text{KMOD}) + 1 . \qquad (6)$$

If that location has previously been selected for some other point, the first available location following I is used instead. The row dimension of INV is approximately 2*MAXGRD, where MAXGRD is the upper bound on the number of grid points.

Subroutine BRIGIT is called after all connection cards have been read and the connection table is complete. Prior to the call, the internal numbers were assigned to the original grid numbers in order of their occurrence. BRIGIT performs a sort of the original numbers, assigns new sequential internal numbers, and converts the connection table IG and the array NORIG to the new set of internal numbers. Here, NORIG(I) contains the original grid point number corresponding to the internal number I.

Subroutine SORT sorts a list of length NL. The routine operates fastest on those lists not badly out of order.

Subroutine SETIG makes additions to the connection table IG. For example, if KG1 and KG2 are two connected grid points, this routine sets IG(KG1, J) = KG2 and IG(KG2, K) = KG1, for some J and K, if this connection has not already been set.

Subroutine TIGER makes additions to the connection table IG required by the presence of MPC's. The dependent grid points (those appearing first in each equation) are stored in array LIST for later removal from IG. This routine is called by NASNUM and executed only if the user elects to take the MPC's into account by inserting the card $MPC YES into the NASTRAN deck.

Subroutine SWITCH generates a new connection table IG according to a correspondence table KT which is set up prior to the call. Here KT(I) contains the new designation to be assigned to the grid number currently labeled I, i.e., KT(old) = new.

Subroutine MORRIS deletes all reference in the connection table IG to those points stored in an array LIST of length NL.

Subroutine FIXIT compresses out zeroes and multiple entries in an array LIST originally of length NL. A corrected length NL is returned

to the calling program.

Subroutine SCHEME is the executive for the actual renumbering strategy. The principal quantities required before the call are the number of nodes NN, an upper bound MM on the maximum nodal degree, and the connection table IG. IG is an NN x MM matrix such that a typical element IG(I, J) contains the label of the J^{th} node adjacent to node I. The node labels referred to here are the permanent (sorted) set of internal numbers assigned by BANDIT.

SCHEME first determines the degree of each node, the most prevalent nodal degree, the number of components, the maximum nodal degree, and the original bandwidth. Then, for each component, a list of starting nodes is supplied by DIAM followed by a resequencing by RELABL for each starting node. The numbering sequence yielding the lowest bandwidth and profile is eventually chosen as the new numbering sequence. The output from SCHEME includes an array ILD, where ILD(I) contains the new label corresponding to the original internal label I.

Subroutine STACK is called by SCHEME after the basic renumbering has been completed. This routine determines all points of zero degree and places them last in the numbering sequence.

Subroutine REVERS reverses the numbering of the first NN-KT grid points, where NN is the total number of grid points and KT is the number of points of zero degree. The variable KT is set in subroutine STACK prior to any calls to REVERS.

Subroutine DEGREE sets up the IDEG array containing in location I the degree of node I.

Function MODE has as its value the most prevalent nodal degree. If several degrees are equally prevalent, the lowest is chosen.

Function COMPNT has as its value the number of components stored in the array IG. This function also sets up arrays IC, in which the I^{th} element contains a component index for the node labeled I, and ICC, in which the I^{th} element contains an index indicating the starting position to be used for labels for component I. Thus, the number of elements in component I is given by

$$ICC(I+1) - ICC(I) .$$

Function MAXDGR has as its value the maximum degree of any node of component $NC > 0$. If the formal parameter $NC \leq 0$, all components are considered.

Function MAXBND has as its value the maximum difference between node labels of connected nodes for nodes of component $NC > 0$. If the parameter $NC \leq 0$, all components are considered and hence the bandwidth is computed. This routine also computes IH, the matrix profile.

Function MINDEG has as its value the minimum degree of any node of component $NC > 0$. If $NC \leq 0$, all components are considered.

Subroutine DIAM determines NL starting nodes and stores the list in the array NODESL.

Subroutine RELABL generates a relabeling scheme starting with NS nodes whose labels are stored in the array NODES. Although this routine allows for multiple starting nodes, BANDIT currently considers only one starting node at a time (corresponding to NS = 1). The relabeling permutation developed by RELABL is stored in ILD and NEW. ILD(I) contains the new label for the node labeled I in the original numbering scheme. The NEW array is the inverse of ILD.

Function IDIST has as its value the maximum distance of any node in component IC(NS) from the node NS. The distance of each node in

this component is stored in the array IDIS. The maximum number of nodes at the same distance from NS is stored in ML.

The COMPASS routine PCUP has three entry points used by BANDIT: PACK, IUNPK, and ABT. The first two are for integer packing and unpacking, respectively. Since most of the BANDIT variables are integer, the CDC 60-bit word length is wasteful of available core. Thus, to reduce the overall core requirements, the connection table IG is packed with four, five, or six integers per word. For example, instead of the FORTRAN statement

$$\text{IG(I, J)} = \text{L} , \tag{7}$$

we have

$$\text{CALL PACK(IG, MAXGRD*(J-1)+I, NBITIN, L).} \tag{8}$$

Similarly, instead of

$$\text{M} = \text{IG(I, J)} , \tag{9}$$

we have

$$\text{M} = \text{IUNPK(IG, MAXGRD*(J-1)+I, NBITIN).} \tag{10}$$

Here, the first three arguments refer to the function name, the location, and the number of bits per integer, respectively, and MAXGRD is the row-dimension of IG. Statements of these types appear throughout the coding of the CDC version of BANDIT.

Subroutine ABT causes an abnormal termination of BANDIT.

The other COMPASS routine, CORSIZ, provides BANDIT with its variable-core feature by interrogating the system during execution to determine the field length (FL) and the distance from the first word of blank common to the end of the FL. These values are returned to the calling program through common block /K/ .

The BANDIT listing in Appendix B was prepared for machines other than CDC machines and omits the routines written in the COMPASS assembly language. Hence this version of BANDIT has no integer packing and is a fixed-core program. Additional changes were made in BANDIT, NASNUM, GRID, and BOMBIT and given new end-punching (JJ).

The following disk files are used by this version of BANDIT: TAPE5 (input), TAPE6 (output), TAPE7 (punch), TAPE9 (scratch), TAPE11 (scratch), and TAPE8. The latter is a BCD file which, after a successful BANDIT run, contains the complete NASTRAN data deck.

The Appendix B version of BANDIT could easily be converted into a variable-core program for any machine for which an assembly language routine could be written to determine the length of blank common storage.

ACKNOWLEDGMENTS

The author wishes to thank Dr. E. H. Cuthill and James M. McKee for ideas resulting from many interesting and stimulating discussions and for the following research subroutines which have been incorporated into BANDIT: SCHEME, DEGREE, COMPNT, MAXDGR, MAXBND, MINDEG, DIAM, RELABL, and IDIST. Appreciation is also expressed to Michael E. Golden for the assembly language routines which he developed.

APPENDIX A
LISTING OF THE CDC VERSION OF BANDIT

```
      PROGRAM BANDIT(INPUT=601,OUTPUT=601,PUNCH=161,INSERT=161,          BANDIT 2      1
     1 TAPE5=INPUT,TAPE6=OUTPUT,TAPE7=PUNCH,TAPE12=INSERT,               BANDIT 3      2
     2 TAPE8=801,TAPE9=801,TAPE11=161)                                   BANDIT 4      3
C                                                                        BANDIT 5      4
C                    B A N D I T                                         BANDIT 6      5
C                                                                        BANDIT 7      6
C MAIN PROGRAM FOR THE RENUMBERING OF NASTRAN GRID POINTS FOR            BANDIT 8      7
C      REDUCED BANDWIDTH.                                                BANDIT 9      8
C THE NASTRAN DATA DECK MUST CONTAIN A BEGIN BULK CARD IN ITS            BANDIT10      9
C      PROPER PLACE AND TERMINATE WITH AN ENDDATA CARD.                  BANDIT11     10
      DIMENSION A(20)                                                    BANDIT12     11
      COMMON KOM(1000)                                                   BANDIT13     12
      COMMON /S/ NN,MM,IH,IB                                             BANDIT14     13
      COMMON /P/ IH0,IHE                                                 BANDIT15     14
      COMMON /A/ MAXGRD,MAXDEG,KMOD,NMPC                                 BANDIT16     15
      COMMON /B/ IPARAM(20),IARG(5)                                      BANDIT17     16
      COMMON /C/ IWARN,LINE,KORIG,KNEW                                   BANDIT18     17
      COMMON /K/ II(7),KORE,IFL                                          BANDIT19     18
      COMMON /BITS/ NBITIN,NBITEX,IPASS                                  BANDIT20     19
      COMMON /TIME/ STIME,NCOMP                                          BANDIT21     20
      COMMON /NEL/ NEL,TIM2                                              BANDIT22     21
      COMMON /DOL/ ISTART(100),IGNORE(100),IFIRST(100)                   BANDIT23     22
      COMMON /DOLL/ IDIM,ISTA,IIG,IFIR,IGDEG,ISCH                        BANDIT24     23
      COMMON /ZERO/ KT                                                   BANDIT25     24
      COMMON /NG/ NGRID,CLOCK                                            BANDIT26     25
      INTEGER EOF                                                        BANDIT27     26
      DATA BEGI,ENDD,SEQG/4HBEGI,4HENDD,4HSEQG/                          BANDIT28     27
      CALL SECOND(TIM1)                                                  BANDIT29     28
      CALL REMARK(40H== N A S T R A N    --    B A N D I T ==)           BANDIT30     29
      CALL DATE(DAY)                                                     BANDIT31     30
      CALL TIME(CLOCK)                                                   BANDIT32     31
C DETERMINE KORE, THE DIMENSION OF THE KOM ARRAY, AND IFL, THE FL.       BANDIT33     32
      CALL CORSIZ                                                        BANDIT34     33
C SET NGRID DEFAULT.                                                     BANDIT35     34
      NGRID=KORE/12                                                      BANDIT36     35
      IF(NGRID.GT.2045) NGRID=KORE/13                                    BANDIT37     36
C SET SCHEME DEFAULTS.                                                   BANDIT38     37
      IARG(1)=80                                                         BANDIT39     38
      IARG(2)=1                                                          BANDIT40     39
      IARG(3)=2                                                          BANDIT41     40
      IARG(4)=2                                                          BANDIT42     41
      IARG(5)=0                                                          BANDIT43     42
C SET NUMBER OF BITS PER WORD FOR INTERNAL AND EXTERNAL                  BANDIT44     43
C          GRID NUMBERS.                                                 BANDIT45     44
      NBITIN=12                                                          BANDIT46     45
      NBITEX=60                                                          BANDIT47     46
    7 FORMAT(1H1,16(/),                                                  BANDIT48     47
     1 36X,57HBBBBBB      AAAAA    N     N   DDDDDD    IIIIIII   TTTTTTT/BANDIT49     48
     2 36X,57HB     B    A   A    NN    N   D     D      I         T   /BANDIT50     49
     3 36X,57HB     B    A   A    N N   N   D     D      I         T   /BANDIT51     50
     4 36X,57HBBBBBB     A   A    N  N  N   D     D      I         T   /BANDIT52     51
     5 36X,57HB     B    AAAAAAA  N   N N   D     D      I         T   /BANDIT53     52
     6 36X,57HB     B    A   A    N    NN   D     D      I         T   /BANDIT54     53
     7 36X,57HBBBBBB     A   A    N     N   DDDDDD    IIIIIII     T   )BANDIT55     54
    8 FORMAT(22(/),48X,34HTHEORY OF STRUCTURES BRANCH (1844) /            BANDIT56     55
     1      46X,38HCOMPUTATION AND MATHEMATICS DEPARTMENT  /              BANDIT57     56
     2      44X,42HNAVAL SHIP RESEARCH AND DEVELOPMENT CENTER /           BANDIT58     57
     3  53X,24HBETHESDA, MARYLAND 20034 )                                 BANDIT59     58
    9 FORMAT(/61X,8HCDC 6700/57X,                                        BANDIT60     59
     +  16HREV. 10 MAR 1972 )                                            BANDIT61     60
   10 FORMAT(20A4)                                                       BANDIT62     61
   11 FORMAT(1H ,20A4)                                                   BANDIT63     62
   12 FORMAT(1H1)                                                        BANDIT64     63
   13 FORMAT(///26H TOTAL CP TIME IN BANDIT = ,F9.3,6H  SEC.)            BANDIT65     64
      LINE=55                                                            BANDIT66     65
      KNEW=0                                                             BANDIT67     66
      REWIND 8                                                           BANDIT68     67
C PRINT TITLE PAGE.                                                      BANDIT69     68
      WRITE(6,7)                                                         BANDIT70     69
      WRITE(6,8)                                                         BANDIT71     70
      WRITE(6,9)                                                         BANDIT72     71
C INITIALIZE VARIABLES.                                                  BANDIT73     72
      DO 15 J=1,20                                                       BANDIT74     73
   15 IPARAM(J)=0                                                        BANDIT75     74
      IPARAM(12)=4                                                       BANDIT76     75
      IDIM=100                                                           BANDIT77     76
      ISTA=0                                                             BANDIT78     77
      IIG=0                                                              BANDIT79     78
      ISCH=0                                                             BANDIT80     79
      IFIR=0                                                             BANDIT81     80
      IGDEG=0                                                            BANDIT82     81
      DO 18 I=1,IDIM                                                     BANDIT83     82
      ISTART(I)=0                                                        BANDIT84     83
      IFIRST(I)=0                                                        BANDIT85     84
   18 IGNORE(I)=0                                                        BANDIT86     85
      IPASS=0                                                            BANDIT87     86
      NN=0                                                               BANDIT88     87
      MM=0                                                               BANDIT89     88
      MAXGRD=0                                                           BANDIT90     89
      MAXDEG=0                                                           BANDIT91     90
      KMOD=0                                                             BANDIT92     91
      KORIG=0                                                            BANDIT93     92
      KNEW=0                                                             BANDIT94     93
      STIME=0.                                                           BANDIT95     94
      NCOMP=0                                                            BANDIT96     95
      NEL=0                                                              BANDIT97     96
      KT=0                                                               BANDIT98     97
      TIM2=0.                                                            BANDIT99     98
      REWIND 9                                                           BANDI100     99
C READ DECK FOR FIRST TIME.                                              BANDI101    100
```

```
      CALL GOOGAN(1,2,5,9)                                              BANDI102   101
C SLICE UP CORE ACCORDING TO SUBROUTINE GRID.                           BANDI103   102
      K2=II(1)*II(2)+1                                                  BANDI104   103
      K3=K2+II(3)*2                                                     BANDI105   104
      K4=K3+II(4)                                                       BANDI106   105
      K5=K4+II(5)                                                       BANDI107   106
      K6=K5+II(6)                                                       BANDI108   107
      K7=K6+II(7)                                                       BANDI109   108
C PROCESS DECK.                                                         BANDI110   109
      CALL NASNUM(KOM(1),II(1),KOM(K2),II(3),KOM(K3),KOM(K4),KOM(K5),   BANDI111   110
     + KOM(K6),KOM(K7),KOM(1),KOM(K2))                                  BANDI112   111
C      ARRAY STARTING AT LOCATION K7 HAS DIMENSION 2*MAXDEG             BANDI113   112
C PROCESS OUTPUT ACCORDING TO OUTPUT REQUESTS.                          BANDI114   113
C CHECK IF CONNECTION CARDS IN DECK.                                    BANDI115   114
      IF(IPARAM(9).EQ.3)GO TO 19                                        BANDI116   115
      REWIND 8                                                          BANDI117   116
      REWIND 9                                                          BANDI118   117
      FLAG=ENDD                                                         BANDI119   118
      J=0                                                               BANDI120   119
      K=9                                                               BANDI121   120
      GO TO 20                                                          BANDI122   121
   19 REWIND 8                                                          BANDI123   122
      J=0                                                               BANDI124   123
      K=8                                                               BANDI125   124
      FLAG=ENDD                                                         BANDI126   125
      IF(IPARAM(5).EQ.4)GO TO 20                                        BANDI127   126
      K=9                                                               BANDI128   127
      IF(IPARAM(6).EQ.3)FLAG=BEGI                                       BANDI129   128
   20 READ(K,10)A                                                       BANDI130   129
      IF(EOF(K).NE.0)CALL BOMBIT(1)                                     BANDI131   130
      J=J+1                                                             BANDI132   131
      IF(IPARAM(10).EQ.5.AND.A(1).NE.SEQG) J=J-1                        BANDI133   132
      IF(MOD(J,LINE).EQ.1)WRITE(6,12)                                   BANDI134   133
      IF(IPARAM(10).EQ.6) WRITE(6,11) A                                 BANDI135   134
      IF(IPARAM(10).EQ.5.AND.A(1).EQ.SEQG) WRITE(6,11)A                 BANDI136   135
      IF(IPARAM(1).EQ.2)WRITE(7,10)A                                    BANDI137   136
      IF(IPARAM(1).EQ.1.AND.A(1).EQ.SEQG) WRITE(7,10) A                 BANDI138   137
      IF(K.NE.8) WRITE(8,10) A                                          BANDI139   138
      IF(A(1).NE.FLAG)GO TO 20                                          BANDI140   139
      IF(FLAG.EQ.ENDD)GO TO 25                                          BANDI141   140
      FLAG=ENDD                                                         BANDI142   141
      K=5                                                               BANDI143   142
      GO TO 20                                                          BANDI144   143
   25 CALL SECOND(TIM2)                                                 BANDI145   144
      TIM2=TIM2-TIM1                                                    BANDI146   145
      IF(IPARAM(5).EQ.3)GO TO 60                                        BANDI147   146
      IF(IPARAM(7).EQ.4)GO TO 60                                        BANDI148   147
      IF(IPARAM(9).EQ.4)GO TO 60                                        BANDI149   148
C USER SUMMARY.                                                         BANDI150   149
      WRITE(6,50) KORIG,KNEW,TIM2                                       BANDI151   150
   50 FORMAT(23H1***BANDIT USER SUMMARY /                               BANDI152   151
     1 8X,25HORIGINAL SEMI-BANDWIDTH = ,I9/                             BANDI153   152
     2 8X,20HNEW SEMI-BANDWIDTH = ,I14/                                 BANDI154   153
     3    8X,19HCP TIME IN BANDIT = ,F9.3,6H  SEC. )                    BANDI155   154
      WRITE(6,117) IH0,IHE                                              BANDI156   155
  117 FORMAT(8X,18HORIGINAL PROFILE =,I16/8X,13HNEW PROFILE =,I21)      BANDI157   156
      WRITE(6,104) NN                                                   BANDI158   157
      WRITE(6,113) NEL                                                  BANDI159   158
      WRITE(6,112) NCOMP                                                BANDI160   159
      WRITE(6,107) MM                                                   BANDI161   160
  107 FORMAT(8X,22HMAXIMUM NODAL DEGREE = ,I12)                         BANDI162   161
      WRITE(6,116) KT                                                   BANDI163   162
      I=IPARAM(1)                                                       BANDI164   163
      IF(I.EQ.1) WRITE(6,101)                                           BANDI165   164
      IF(I.EQ.2) WRITE(6,102)                                           BANDI166   165
      IF(I.EQ.3) WRITE(6,103)                                           BANDI167   166
  101 FORMAT(8X,34HPUNCH OUTPUT            SEQGP CARDS )                BANDI168   167
  102 FORMAT(8X,34HPUNCH OUTPUT              ALL CARDS )                BANDI169   168
  103 FORMAT(8X,34HPUNCH OUTPUT                   NONE )                BANDI170   169
      WRITE(6,119) IFL                                                  BANDI171   170
      WRITE(6,105) MAXGRD,MAXDEG                                        BANDI172   171
  105 FORMAT(18X,8HMAXGRD =,I11/18X,8HMAXDEG = ,I11)                    BANDI173   172
      WRITE(6,109) KORE,KORE                                            BANDI174   173
  109 FORMAT(18X,6HKORE =,I13/18X,6HKORE =,6X,O6,1HB)                   BANDI175   174
      WRITE(6,111) DAY,CLOCK                                            BANDI176   175
  111 FORMAT(8X,14HDATE AND TIME ,2A10)                                 BANDI177   176
C IPASS=NUMBER OF PCUP CALLS.                                           BANDI178   177
  104 FORMAT(8X,23HNUMBER OF GRID POINTS = ,I11)                        BANDI179   178
  113 FORMAT(8X,20HNUMBER OF ELEMENTS = ,I14)                           BANDI180   179
  112 FORMAT(8X,22HNUMBER OF COMPONENTS = ,I12)                         BANDI181   180
  116 FORMAT(8X,28H# OF POINTS OF ZERO DEGREE = ,I6)                    BANDI182   181
  119 FORMAT(8X,19HFIELD LENGTH (FL) = ,8X,O6,1HB)                      BANDI183   182
      GO TO 70                                                          BANDI184   183
   60 IF(IPARAM(10).EQ.5) WRITE(6,12)                                   BANDI185   184
      WRITE(6,13) TIM2                                                  BANDI186   185
   70 CONTINUE                                                          BANDI187   186
      REWIND 8                                                          BANDI188   187
      IF(IPARAM(8).EQ.4) STOP 5                                         BANDI190   188
      STOP                                                              BANDI191   189
      END                                                               BANDI192   190
      SUBROUTINE NASNUM(IG,II1,INV,II3,INT,ICC,ILD,NORIG,IP,JG,JNV)     NASNUM 2   191
      DIMENSION A(8),KG(40),LG(40),LINE(10),B(20),ATEMP(4)              NASNUM 3   192
      DIMENSION IG(II1,1),INV(II3,2),JG(1),JNV(1)                       NASNUM 4   193
      DIMENSION INT(1),ICC(1),ILD(1),NORIG(1),IP(1)                     NASNUM 5   194
C IP HAS DIMENSION 2*MAXDEG.   JG AND JNV ARE EQUIV TO IG AND INV.      NASNUM 6   195
      COMMON /S/ NN,MM,IH,IB                                            NASNUM 7   196
      COMMON /A/ MAXGRD,MAXDEG,KMOD,NMPC                                NASNUM 8   197
      COMMON /B/ IPARAM(20),IARG(5)                                     NASNUM 9   198
      COMMON /C/ IWARN,NLINE,KORIG,KNEW                                 NASNUM10   199
      COMMON /BITS/ NBITIN,NBITEX,IPASS                                 NASNUM11   200
```

```
      COMMON /K/ II(7),KORE                                              NASNUM12   201
      COMMON /TIME/ TIM2,NCOMP                                           NASNUM13   202
      COMMON /NEL/ NEL                                                   NASNUM14   203
      COMMON /DOL/ ISTART(100),IGNORF(100),IFIRST(100)                   NASNUM15   204
      COMMON /DOLL/ IOIM,ISTA,IIG,IFIR,IGOEG,ISCH                        NASNUM16   205
C THE VARIABLE LINE DEFINED NEAR CARD NASNUM.300 IS NOT THE              NASNUM17   206
C       SAME AS THE VARIABLE LINE APPEARING IN COMMON                    NASNUM18   207
C         IN OTHER ROUTINES.                                             NASNUM19   208
      DIMENSION TYPE(50),WYPE(50)                                        NASNUM20   209
      DIMENSION F1A(2),F10A(2),F1B(2),F10B(2)                            NASNUM21   210
      DATA BEGI,ENDD,SEQG/4HBEGI,4HENDD,4HSEQG/                          NASNUM22   211
      DATA TYPE/4HCBAR,4HCELA,4HCELA,4HCONR,4HCQDM,4HCQDP,4HCQUA,        NASNUM23   212
     1   4HCQUA,4HCQUA,4HCROD,4HCSHE,4HCTRB,4HCTRI,4HCTRI,4HCTRM,        NASNUM24   213
     2   4HCTRP,4HCTUB,4HCTWI,4HENDD,4HMPC*,4HCDAM,4HCDAM,4HCMAS,        NASNUM25   214
     3   4HCMAS,4HCVIS,4HCDAM,4HCDAM,4HCELA,4HCELA,4HCMAS,4HCMAS,        NASNUM26   215
     4   4HCCON,4HCTOR,4HCTRA,4HCTRI,4HCONM,4HCONM,4HCHTT,4HCIS3,        NASNUM27   216
     5   4HCIS3,4HCIS2,4HCIS2,4HCISH,4HCISH,4HCFLU,4HCFLU,4HCFLU,        NASNUM28   217
     6   4HCIET,4HCHEX,4HCHEX/                                          NASNUM29   218
      DATA WYPE/4H*    ,4HS1* ,4HS2* ,4HOD* ,4HEM* ,4HLT* ,4HD1* ,      NASNUM30   219
     1   4HD2* ,4HD3* ,4H*    ,4HAR* ,4HSC* ,4HA1* ,4HA2* ,4HEM* ,       NASNUM31   220
     2   4HLT* ,4HE*  ,4HST* ,4HATA ,4H     ,4HP1* ,4HP2* ,4HS1* ,       NASNUM32   221
     3   4HS2* ,4HC*  ,4HP3* ,4HP4* ,4HS3* ,4HS4* ,4HS3* ,4HS4* ,       NASNUM33   222
     4   4HEAX*,4HORG*,4HPRG*,4HARG*,4H1*  ,4H2*  ,4HRI2*,4HD8* ,       NASNUM34   223
     5   4HD20*,4HD4* ,4HD8* ,4H8*  ,4H16* ,4HID2*,4HID3*,4HID4*,       NASNUM35   224
     6   4HRA* ,4HA1* ,4HA2* /                                          NASNUM36   225
      NTYPE=50                                                           NASNUM37   226
      REWIND 8                                                           NASNUM38   227
      REWIND 9                                                           NASNUM39   228
      NMPC=40                                                            NASNUM40   229
      KMOD=2.*FLOAT(MAXGRD)-2.2715*SQRT(1.131*FLOAT(MAXGRD))             NASNUM41   230
      NEW=0                                                              NASNUM42   231
      IWARN=0                                                            NASNUM43   232
      NEQ=0                                                              NASNUM44   233
    2 FORMAT(29H1BANDIT INFORMATION MESSAGE -                            NASNUM45   234
     +19H      NO GRID POINTS/                                           NASNUM46   235
     +28H      RESEQUENCING SUPPRESSED)                                  NASNUM47   236
    4 FORMAT(    19H ***NEW BANDWIDTH =,I6)                              NASNUM48   237
    5 FORMAT(33H1THE WRONG CARD FOLLOWS THIS CARD/1X,2A4,1P4E16.7,2A4// NASNUM49   238
     2       40H THE CONTINUATION CARD IS REQUIRED NEXT ,                NASNUM50   239
     3      36HSINCE BANDIT DOES NOT SORT THE DECK.                      NASNUM51   240
     4 13H FATAL ERROR. )                                                NASNUM52   241
    6 FORMAT(1H1)                                                        NASNUM53   242
    7 FORMAT(54H1 ONE OR MORE SEQGP CARDS ALREADY APPEAR IN DATA DECK./  NASNUM54   243
     +  55H  RESEQUENCING CANNOT BE REQUESTED.        FATAL ERROR. )     NASNUM55   244
    8 FORMAT(5HSEQGP,3X,2I8,56X)                                         NASNUM56   245
    9 FORMAT(20A4)                                                       NASNUM57   246
   10 FORMAT(2A4,4F16.0,2A4)                                             NASNUM58   247
   11 FORMAT(1H ,5(I8,I11,7X))                                           NASNUM59   248
   12 FORMAT(5HSFQGP,3X,8I8,8X)                                          NASNUM60   249
   14 FORMAT(////26H ***BANDIT WARNING MESSAGE /                         NASNUM61   250
     1 11X,35HTHE WRONG CARD MAY FOLLOW THIS CARD /                      NASNUM62   251
     2  11X,2A4,1P4E16.7,2A4/                                            NASNUM63   252
     3      11X,47HCHECK INPUT DECK TO BE SURE THAT A CONTINUATION ,     NASNUM64   253
     4  42H CARD IS NEITHER MISSING NOR OUT OF SORT.   )                 NASNUM65   254
   15 FORMAT(26H TOTAL CP TIME IN SCHEME = ,F9.3,6H  SEC. )              NASNUM66   255
   19 FORMAT(1H1,5(20HINTERNAL    ORIGINAL,6X)/                         NASNUM67   256
     11H ,5(20HGPID NO.    GRID PT.,6X))                                NASNUM68   257
C RETURN IF RESEQUENCING IS NOT DESIRED.                                 NASNUM69   258
      IF(IPARAM(5).EQ.3)RETURN                                           NASNUM70   259
C CHECK IF SEQGP CARDS ALREADY APPEAR IN DECK.                           NASNUM71   260
      IF(IPARAM(7).EQ.3)GO TO 22                                         NASNUM72   261
C ABORT BANDIT SINCE SEQGP CARDS ALREADY APPEAR IN DECK.                 NASNUM73   262
      WRITE(6,7)                                                         NASNUM74   263
      CALL BOMBIT(3)                                                     NASNUM75   264
C READ AND EXTRACT CONNECTION CARDS FROM DECK.                           NASNUM76   265
   22 CALL GOOGAN(2,1,9,8)                                               NASNUM77   266
      REWIND 8                                                           NASNUM78   267
      REWIND 9                                                           NASNUM79   268
C INITIALIZE EXPANDABLE CORE.                                            NASNUM80   269
      DO 30 I=1,KORE                                                     NASNUM81   270
   30 JG(I)=0                                                            NASNUM82   271
C READ CARD.                                                             NASNUM83   272
   40 READ(8,10)F1A,(A(I),I=1,4),F10A                                    NASNUM84   273
C DETERMINE CARD TYPE.                                                   NASNUM85   274
   45 ITYPE=0                                                            NASNUM86   275
      DO 50 I=1,NTYPE                                                    NASNUM87   276
   50 IF(F1A(1).EQ.TYPE(I).AND.F1A(2).EQ.WYPE(I)) ITYPE=I                NASNUM88   277
      IF(ITYPE.EQ.0)GO TO 40                                             NASNUM89   278
      IF(ITYPE.EQ.19)GO TO 500                                           NASNUM90   279
      IF(ITYPE.EQ.20.AND.IPARAM(4).EQ.3)GO TO 40                         NASNUM91   280
C READ CONTINUATION TO CARD JUST READ.                                   NASNUM92   281
      READ(8,10)F1B,(A(I),I=5,8),F10B                                    NASNUM93   282
C CHECK EACH LOGICAL CARD FOR PROPER SORT.                               NASNUM94   283
      IF(F1B(1).EQ.F10A(1).AND.F1B(2).EQ.F10A(2)) GO TO 60               NASNUM95   284
C--- IF FOLLOWING CARD TYPES ARE OUT OF SORT, NO ERROR                   NASNUM96   285
      IF(ITYPE.EQ.1.OR.ITYPE.EQ.4)GO TO 56                               NASNUM97   286
      IF(ITYPE.EQ.32)GO TO 56                                            NASNUM98   287
      IF(ITYPE.EQ.33)GO TO 56                                            NASNUM99   288
      IF(ITYPE.EQ.35)GO TO 56                                            NASNU100   289
      IF(ITYPE.EQ.36)GO TO 56                                            NASNU101   290
      IF(ITYPE.EQ.37)GO TO 56                                            NASNU102   291
      IF(ITYPE.EQ.45.OR.ITYPE.EQ.46) GO TO 56                            NASNU103   292
C--- IF FOLLOWING CARD TYPES ARE OUT OF SORT, POSSIBLE ERROR (GIVE       NASNU104   293
C---            WARNING MESSAGE)                                         NASNU105   294
      IWARN=IWARN+1                                                      NASNU106   295
      IF(MOD(IWARN,6).EQ.1)WRITE(6,6)                                    NASNU107   296
      IF(ITYPE.EQ. 2) GO TO 54                                           NASNU108   297
      IF(ITYPE.EQ. 3) GO TO 54                                           NASNU109   298
      IF(ITYPE.EQ.10) GO TO 54                                           NASNU110   299
      IF(ITYPE.EQ.17) GO TO 54                                           NASNU111   300
```

```
      IF(ITYPE.GE.21.AND.ITYPE.LE.31)GO TO 54                          NASNU112   301
C--- FOR OTHER CARD TYPES OUT OF SORT, ABORT BANDIT                     NASNU113   302
   52 WRITE(6,5)F1A,(A(I),I=1,4),F10A                                   NASNU114   303
      CALL JOMBIT(2)                                                    NASNU115   304
   54 WRITE(6,14)F1A,(A(I),I=1,4),F10A                                  NASNU116   305
C SAVE CONTENTS OF THE SECOND CARD OF THE PAIR.                         NASNU117   306
   55 DO 58 I=1,4                                                       NASNU118   307
      ATEMP(I)=A(I+4)                                                   NASNU119   308
   58 A(I+4)=0.                                                         NASNU120   309
C INITIALIZE KG AND LG.                                                 NASNU121   310
   60 DO 70 I=1,NMPC                                                    NASNU122   311
      KG(I)=0                                                           NASNU123   312
   70 LG(I)=0                                                           NASNU124   313
      LOOP=1                                                            NASNU125   314
      NCON=4                                                            NASNU126   315
C SET UP KG AND LG.  **                                                 NASNU127   316
      GO TO (160,220,220,200,120,120,120,120,120,180,120,140,140,       NASNU128   317
     1       140,140,140,180,120,500,230,220,220,220,220,180,180,       NASNU129   318
     2       180,180,180,180,180,160,160,110,114,118,118,140, 80,       NASNU130   319
     3        95,120, 90, 80, 95,200,114,110,120, 90, 90),ITYPE          NASNU131   320
C* CIS3D8,CISH8                                                         NASNU132   321
   80 DO 81 I=1,7                                                       NASNU133   322
   81 KG(I)=A(I+1)+0.5                                                  NASNU134   323
      NCON=8                                                            NASNU135   324
      READ(8,10) F1A,A(1),A(2),A(3),A(4),F10A                           NASNU136   325
      IF(F1A(1).NE.F10B(1).OR.F1A(2).NE.F10B(2)) GO TO 100              NASNU137   326
      KG(8)=A(1)+0.5                                                    NASNU138   327
      GO TO 250                                                         NASNU139   328
C* CIS3D20                                                             NASNU140   329
   85 DO 86 I=1,7                                                       NASNU141   330
   86 KG(I)=A(I+1)+0.5                                                  NASNU142   331
      NCON=20                                                           NASNU143   332
      READ(8,10) F1A,A(1),A(2),A(3),A(4),F10A                           NASNU144   333
      IF(F1A(1).NE.F10B(1).OR.F1A(2).NE.F10B(2)) GO TO 100              NASNU145   334
      READ(8,10) F1B,A(5),A(6),A(7),A(8),F10B                           NASNU146   335
      IF(F1B(1).NE.F10A(1).OR.F1B(2).NE.F10A(2)) GO TO 52               NASNU147   336
      DO 87 I=9,15                                                      NASNU148   337
   87 KG(I)=A(I-7)+0.5                                                  NASNU149   338
      READ(8,10) F1A,A(1),A(2),A(3),A(4),F10A                           NASNU150   339
      IF(F1A(1).NE.F10B(1).OR.F1A(2).NE.F10B(2)) GO TO 100              NASNU151   340
      READ(8,10) F1B,A(5),A(6),A(7),A(8),F10B                           NASNU152   341
      IF(F1B(1).NE.F10A(1).OR.F1B(2).NE.F10A(2)) GO TO 52               NASNU153   342
      DO 88 I=16,20                                                     NASNU154   343
   88 KG(I)=A(I-15)+0.5                                                 NASNU155   344
      GO TO 250                                                         NASNU156   345
C* CIS2D8,CHEXA1,CHEXA2                                                 NASNU157   346
   90 DO 91 I=1,6                                                       NASNU158   347
   91 KG(I)=A(I+2)+0.5                                                  NASNU159   348
      NCON=8                                                            NASNU160   349
      READ(8,10) F1A,A(1),A(2),A(3),A(4),F10A                           NASNU161   350
      IF(F1A(1).NE.F10B(1).OR.F1A(2).NE.F10B(2)) GO TO 100              NASNU162   351
      DO 92 I=7,8                                                       NASNU163   352
   92 KG(I)=A(I-6)+0.5                                                  NASNU164   353
      GO TO 250                                                         NASNU165   354
C* CISH16                                                              NASNU166   355
   95 DO 96 I=1,7                                                       NASNU167   356
   96 KG(I)=A(I+1)+0.5                                                  NASNU168   357
      NCON=16                                                           NASNU169   358
      READ(8,10) F1A,A(1),A(2),A(3),A(4),F10A                           NASNU170   359
      IF(F1A(1).NE.F10B(1).OR.F1A(2).NE.F10B(2)) GO TO 100              NASNU171   360
      READ(8,10) F1B,A(5),A(6),A(7),A(8),F10B                           NASNU172   361
      IF(F1B(1).NE.F10A(1).OR.F1B(2).NE.F10A(2)) GO TO 52               NASNU173   362
      DO 97 I=9,15                                                      NASNU174   363
   97 KG(I)=A(I-7)+0.5                                                  NASNU175   364
      READ(8,10) F1A,A(1),A(2),A(3),A(4),F10A                           NASNU176   365
      IF(F1A(1).NE.F10B(1).OR.F1A(2).NE.F10B(2)) GO TO 100              NASNU177   366
      KG(16)=A(1)+0.5                                                   NASNU178   367
      GO TO 250                                                         NASNU179   368
  100 F1A(1)=F1B(1)                                                     NASNU180   369
      F1A(2)=F1B(2)                                                     NASNU181   370
       DO 101 I=1,4                                                     NASNU182   371
  101 A(I)=A(I+4)                                                       NASNU183   372
      F10A(1)=F10B(1)                                                   NASNU184   373
      F10A(2)=F10B(2)                                                   NASNU185   374
      GO TO 52                                                          NASNU186   375
C* CTRAPRG,CFLUID4                                                      NASNU187   376
  110 DO 112 I=1,4                                                      NASNU188   377
  112 KG(I)=A(I+1)+0.5                                                  NASNU189   378
      GO TO 250                                                         NASNU190   379
C* CTRIARG,CFLUID3                                                      NASNU191   380
  114 DO 116 I=1,3                                                      NASNU192   381
  116 KG(I)=A(I+1)+0.5                                                  NASNU193   382
      GO TO 250                                                         NASNU194   383
C* CONM1, CONM2                                                         NASNU195   384
  118 KG(1)=A(2)+0.5                                                    NASNU196   385
      KG(2)=KG(1)                                                       NASNU197   386
      GO TO 250                                                         NASNU198   387
C* CQDMEM,CQDPLT,CQUAD1,CQUAD2,CQUAD3,CSHEAR,CTWIST,CIS2D4,CTETRA       NASNU199   388
  120 DO 130 I=1,4                                                      NASNU200   389
  130 KG(I)=A(I+2)+0.5                                                  NASNU201   390
      GO TO 250                                                         NASNU202   391
C* CTRBSC, CTRIA1, CTRIA2, CTRMEM, CTRPLT, CTTRIA2                      NASNU203   392
  140 DO 150 I=1,3                                                      NASNU204   393
  150 KG(I)=A(I+2)+0.5                                                  NASNU205   394
      GO TO 250                                                         NASNU206   395
C* CBAR, CCONEAX, CTORDRG                                               NASNU207   396
  160 DO 170 I=1,2                                                      NASNU208   397
  170 KG(I)=A(I+2)+0.5                                                  NASNU209   398
      GO TO 250                                                         NASNU210   399
C* CROD, CTUBE, CVISC, CDAMP3, CDAMP4, CELAS3, CELAS4, CMASS3, CMASS4   NASNU211   400
```

```
  180 DO 190 I=1,2                                                    NASNU212   401
      KG(I)=A(I+2)+0.5                                                NASNU213   402
  190 LG(I)=A(I+6)+0.5                                                NASNU214   403
C SET LOOP=2 SINCE 2 ELEMENTS MAY BE DEFINED ON ONE CARD.             NASNU215   404
      LOOP=2                                                          NASNU216   405
      GO TO 250                                                       NASNU217   406
C* CONROD,CFLUID2                                                     NASNU218   407
  200 DO 210 I=1,2                                                    NASNU219   408
  210 KG(I)=A(I+1)+0.5                                                NASNU220   409
      GO TO 250                                                       NASNU221   410
C* CELAS1, CELAS2, CDAMP1, CDAMP2, CMASS1, CMASS2                     NASNU222   411
  220 KG(1)=A(3)+0.5                                                  NASNU223   412
      KG(2)=A(5)+0.5                                                  NASNU224   413
      GO TO 250                                                       NASNU225   414
C PROCESS MPC CARDS.                                                  NASNU226   415
  230 NCON=NMPC                                                       NASNU227   416
      KG(1)=A(2)+0.5                                                  NASNU228   417
      KG(2)=A(5)+0.5                                                  NASNU229   418
      I=2                                                             NASNU230   419
  240 READ(8,10)F1A,(A(J),J=1,4),F10A                                 NASNU231   420
      IF(F10B(1).NE.F1A(1).OR.F10B(2).NE.F1A(2)) GO TO 250            NASNU232   421
      I=I+1                                                           NASNU233   422
      IF(I.GT.NMPC)GO TO 245                                          NASNU234   423
      F10B(1)=F10A(1)                                                 NASNU235   424
      F10B(2)=F10A(2)                                                 NASNU236   425
      KK=2                                                            NASNU237   426
      IF(MOD(I,2).EQ.0)KK=1                                           NASNU238   427
      KG(I)=A(KK)+0.5                                                 NASNU239   428
      GO TO 240                                                       NASNU240   429
  245 WRITE(6,246) NMPC                                               NASNU241   430
  246 FORMAT(36H1 AN MPC EQUATION CONTAINS MORE THAN,I5,8H  TERMS./   NASNU242   431
     + 14H  FATAL ERROR. )                                            NASNU243   432
      CALL BOMBIT(5)                                                  NASNU244   433
C PROCESS KG (AND LG IF LOOP=2) ARRAY.                                NASNU245   434
  250 DO 480 KK=1,LOOP                                                NASNU246   435
      IF(KK.EQ.1)GO TO 300                                            NASNU247   436
      DO 260 I=1,4                                                    NASNU248   437
  260 KG(I)=LG(I)                                                     NASNU249   438
C SCATTER SEARCH AND CONVERT KG TO TEMPORARY SET OF INTERNAL NUMBERS.  NASNU250   439
  300 CALL SCAT(KG,NCON,NEW,INV,II3,NORIG)                            NASNU251   440
      IF(ITYPE.NE.20)GO TO 420                                        NASNU252   441
C SAVE MPC GRID POINTS FOR LATER PROCESSING BY TIGER.                 NASNU253   442
      NEQ=NEQ+1                                                       NASNU254   443
      WRITE(11)KG                                                     NASNU255   444
      GO TO 45                                                        NASNU256   445
C FILL CONNECTION TABLE ARRAY IG.                                     NASNU257   446
  420 IEND=NCON-1                                                     NASNU258   447
      NEL=NEL+1                                                       NASNU259   448
      DO 450 I=1,IEND                                                 NASNU260   449
      L=I+1                                                           NASNU261   450
      DO 450 J=L,NCON                                                 NASNU262   451
  450  CALL SETIG(KG(I),KG(J),IG,II1,NORIG)                           NASNU263   452
  480 CONTINUE                                                        NASNU264   453
      IF(F1B(1).EQ.F10A(1).AND.F1B(2).EQ.F10A(2)) GO TO 40            NASNU265   454
      IF(NCON.GE.8) GO TO 40                                          NASNU266   455
      F1A(1)=F1B(1)                                                   NASNU267   456
      F1A(2)=F1B(2)                                                   NASNU268   457
      DO 495 I=1,4                                                    NASNU269   458
  495 A(I)=ATEMP(I)                                                   NASNU270   459
      F10A(1)=F10B(1)                                                 NASNU271   460
      F10A(2)=F10B(2)                                                 NASNU272   461
      GO TO 45                                                        NASNU273   462
  500 NN=NEW                                                          NASNU274   463
      IF(NEW.GT.0) GO TO 502                                          NASNU275   464
      WRITE(6,2)                                                      NASNU276   465
      IPARAM(9)=4                                                     NASNU277   466
      RETURN                                                          NASNU278   467
  502 IF(IPARAM(4).EQ.3)GO TO 505                                     NASNU279   468
C MODIFY CONNECTION TABLE TO ACCOUNT FOR MPC EQUATIONS.               NASNU280   469
      CALL TIGER(NEQ,IG,II1,ILD,NORIG)                                NASNU281   470
      NDEP=NN                                                         NASNU282   471
      CALL FIXIT(ILD,NDEP)                                            NASNU283   472
C GENERATE NEW IG AND NORIG ARRAYS.                                   NASNU284   473
  505 CALL BRIGIT(IG,II1,INV,II3,INT,ICC,NORIG,IP)                    NASNU285   474
C PRINT INTERNAL/EXTERNAL CORRESPONDENCE TABLE.                       NASNU286   475
      LEN=50                                                          NASNU287   476
      IF(IPARAM(10).EQ.5) GO TO 560                                   NASNU288   477
      J=0                                                             NASNU289   478
  510 WRITE(6,19)                                                     NASNU290   479
  520 J=J+1                                                           NASNU291   480
      KEND=0                                                          NASNU292   481
      DO 530 K=1,9,2                                                  NASNU293   482
      L=J+LEN*(K-1)/2                                                 NASNU294   483
      LINE(K)=L                                                       NASNU295   484
      IF(L.GT.NEW) GO TO 550                                          NASNU296   485
      KEND=K+1                                                        NASNU297   486
  530 LINE(K+1)=NORIG(L)                                              NASNU298   487
  550 CONTINUE                                                        NASNU299   488
      IF(KEND.EQ.0)GO TO 560                                          NASNU300   489
      WRITE(6,11)(LINE(K),K=1,KEND)                                   NASNU301   490
      IF(MOD(J,LEN).NE.0)GO TO 520                                    NASNU302   491
      J=J+4*LEN                                                       NASNU303   492
      IF(J.LT.NEW) GO TO 510                                          NASNU304   493
  560 CONTINUE                                                        NASNU305   494
C CONVERT ISTART,IGNORE,IFIRST FROM ORIGINAL TO INTERNAL NUMBERS.     NASNU306   495
      I=ISTA+IIG+IFIR                                                 NASNU307   496
      IF(I.LE.0) GO TO 570                                            NASNU308   497
      CALL FLIP(ISTART,ISTA,INV,II3,ICC)                              NASNU309   498
      CALL FLIP(IGNORE,IIG ,INV,II3,ICC)                              NASNU310   499
      CALL FLIP(IFIRST,IFIR,INV,II3,ICC)                              NASNU311   500
```

```
      IF(IPARAM(10).EQ.5) GO TO 570                                    NASNU312   501
C PRINT INTERNAL NUMBERS FOR $-CARDS.                                  NASNU313   502
      WRITE(6,561)                                                     NASNU314   503
  561 FORMAT(30H1   $ CARDS (INTERNAL NUMBERS) /)                      NASNU315   504
      IF(ISTA.GT.0) WRITE(6,562) (ISTART(I),I=1,ISTA)                  NASNU316   505
      IF(IIG .GT.0) WRITE(6,564) (IGNORE(I),I=1,IIG )                  NASNU317   506
      IF(IFIR.GT.0) WRITE(6,566) (IFIRST(I),I=1,IFIR)                  NASNU318   507
  562 FORMAT(9H $START  ,20I5/100(9X,20I5/))                           NASNU319   508
  564 FORMAT(9H $IGNORE ,20I5/100(9X,20I5/))                           NASNU320   509
  566 FORMAT(9H $FIRST  ,20I5/100(9X,20I5/))                           NASNU321   510
  570  CONTINUE                                                        NASNU322   511
C SET UP LIST OF POINTS TO IGNORE IN INT ARRAY.                        NASNU323   512
      K=0                                                              NASNU324   513
      IF(IPARAM(4).EQ.3) GO TO 920                                     NASNU325   514
      IF(NDEP.LE.0) GO TO 920                                          NASNU326   515
C  MPC DEPENDENT POINTS FIRST.                                         NASNU327   516
      DO 915 I=1,NDEP                                                  NASNU328   517
      J=ILD(I)                                                         NASNU329   518
      IF(J.LE.0) GO TO 915                                             NASNU330   519
      K=K+1                                                            NASNU331   520
      INT(K)=ICC(J)                                                    NASNU332   521
      IF(K.GE.MAXGRD) CALL FIXIT(INT,K)                                NASNU333   522
  915 CONTINUE                                                         NASNU334   523
  920 IF(IGDEG.LE.0) GO TO 940                                         NASNU335   524
C  GRID POINTS WITH DEGREE.GT.IGDEG SECOND.                            NASNU336   525
      IF(IGDEG.GE.MM) GO TO 940                                        NASNU337   526
      CALL DEGREE(IG,II1,INV)                                          NASNU338   527
C HERE, INV(I)=DEGREE OF GRID POINT I                                  NASNU339   528
      DO 930 I=1,NN                                                    NASNU340   529
      IF(JNV(I).LE.IGDEG) GO TO 930                                    NASNU341   530
      K=K+1                                                            NASNU342   531
      INT(K)=I                                                         NASNU343   532
      IF(K.GE.MAXGRD) CALL FIXIT(INT,K)                                NASNU344   533
  930 CONTINUE                                                         NASNU345   534
  940 IF(IIG.LE.0) GO TO 960                                           NASNU346   535
C  $IGNORE POINTS THIRD.                                               NASNU347   536
      DO 950 I=1,IIG                                                   NASNU348   537
      J=IGNORE(I)                                                      NASNU349   538
      IF(J.LE.0) GO TO 950                                             NASNU350   539
      K=K+1                                                            NASNU351   540
      INT(K)=J                                                         NASNU352   541
      IF(K.GE.MAXGRD) CALL FIXIT(INT,K)                                NASNU353   542
  950 CONTINUE                                                         NASNU354   543
C K=NUMBER OF POINTS TO BE IGNORED BEFORE COMPRESSING LIST.            NASNU355   544
  960 IF(K.LE.0) GO TO 970                                             NASNU356   545
C DELETE POINTS LISTED IN INT ARRAY FROM CONNECTION TABLE IG.          NASNU357   546
      CALL MORRIS(INT,K,IG,II1)                                        NASNU358   547
  970 CONTINUE                                                         NASNU359   548
C RENUMBER NODES WITH SUBROUTINE SCHEME.                               NASNU360   549
      IF(IPARAM(10).EQ.6) IARG(5)=1                                    NASNU361   550
      II8=II3/2                                                        NASNU362   551
      CALL SECOND(TIM1)                                                NASNU363   552
      CALL SCHEME(IARG(1),IARG(2),IARG(3),IARG(4),IARG(5),IG,II1,      NASNU364   553
     + JNV(1),JNV(II8+1),JNV(2*II8+1),JNV(3*II8+1),INT,ICC,ILD,IP)     NASNU365   554
      CALL SECOND(TIM2)                                                NASNU366   555
      TIM2=TIM2-TIM1                                                   NASNU367   556
      IF(IPARAM(10).EQ.5) GO TO 580                                    NASNU368   557
      WRITE(6,15)TIM2                                                  NASNU369   558
      WRITE(6,4)I9                                                     NASNU370   559
C WRITE NEW NASTRAN DATA DECK.                                         NASNU371   560
  580 READ(9,9)B                                                       NASNU372   561
      WRITE(8,9)B                                                      NASNU373   562
      IF(B(1).NE.BEGI)GO TO 580                                        NASNU374   563
  590 READ(9,9)B                                                       NASNU375   564
      IF(B(1).GE.SEQG.OR.B(1).EQ.ENDD)GO TO 600                        NASNU376   565
      WRITE(8,9)B                                                      NASNU377   566
      GO TO 590                                                        NASNU378   567
C WRITE SEQGP CARDS.                                                   NASNU379   568
  600 KREM=MOD(NEW,4)                                                  NASNU380   569
      IF(NEW.GE.4) GO TO 605                                           NASNU381   570
      KBEG=1                                                           NASNU382   571
      GO TO 612                                                        NASNU383   572
605   IEND=NEW-KREM-3                                                  NASNU384   573
      DO 610 K=1,IEND,4                                                NASNU385   574
      L=K+3                                                            NASNU386   575
  610 WRITE(8,12) (NORIG(I),ILD(I),I=K,L)                              NASNU387   576
      IF(KREM.EQ.0)GO TO 620                                           NASNU388   577
      KBEG=IEND+4                                                      NASNU389   578
  612 DO 615 I=KBEG,NEW                                                NASNU390   579
  615 WRITE(8,8) NORIG(I),ILD(I)                                       NASNU391   580
C WRITE THE REMAINDER OF THE NASTRAN DECK.                             NASNU392   581
  620 WRITE(8,9)B                                                      NASNU393   582
      IF(B(1).EQ.ENDD)GO TO 700                                        NASNU394   583
      READ(9,9)B                                                       NASNU395   584
      GO TO 620                                                        NASNU396   585
  700 CONTINUE                                                         NASNU397   586
      IF(IPARAM(10).EQ.5) GO TO 900                                    NASNU398   587
C PRINT ORIGINAL GRID POINT CONNECTION TABLE.                          NASNU399   588
      MAXD=MM                                                          NASNU400   589
      L=MAXD/11+1                                                      NASNU401   590
      L=LEN/L                                                          NASNU402   591
705   FORMAT(10H1     GRID,5X,5H  MAX,15X,13H*CONNECTIONS*,5X,         NASNU403   592
     +      23H(ORIGINAL GRID NUMBERS) /5X,                           NASNU404   593
     +      20HPOINT COMP DIST DEGR ,11(8X,1H*) )                     NASNU405   594
710   FORMAT(I10,3I5,11I9/25(25X,11I9/))                               NASNU406   595
      DO 750 I=1,NN                                                    NASNU407   596
      IF(MOD(I,L).EQ.1) WRITE(6,705)                                   NASNU408   597
      DO 720 J=1,MAXD                                                  NASNU409   598
720   IP(J)=0                                                          NASNU410   599
C CALCULATE MOIST AND PRINT TABLE.                                     NASNU411   600
```

```
      MDIST=0                                                          NASNU412    601
      DO 725 J=1,MAXD                                                  NASNU413    602
      K=IUNPK(IG,MAXGRD*(J-1)+I,NBITIN)                                NASNU414    603
      IF(K.EQ.0) GO TO 725                                             NASNU415    604
      MDIST=MAX0(MDIST,IABS(I-K))                                      NASNU416    605
      IP(J)=NORIG(K)                                                   NASNU417    606
  725 CONTINUE                                                         NASNU418    607
      K=NORIG(I)                                                       NASNU419    608
      IP1=INV(I,1)                                                     NASNU420    609
      IP2=INV(MAXGRD+I,1)                                              NASNU421    610
  750 WRITE(6,710) K,IP1,MDIST,IP2,(IP(J),J=1,MAXD)                    NASNU422    611
C PRINT CONNECTION TABLE FOR RENUMBERED NUMBERS.                       NASNU423    612
      DO 780 I=1,NEW                                                   NASNU424    613
  780 ICC(I)=ILO(I)                                                    NASNU425    614
      CALL SWITCH(IG,II1,INT,ICC,IP(1),IP(MAXDEG+1))                   NASNU426    615
      CALL DEGREE(IG,II1,JNV(II8+1))                                   NASNU427    616
      L=COMPNT(IG,II1,JNV(1),JNV(II8+1),JNV(3*II8+1),ICC)              NASNU428    617
      L=MAXD/26+1                                                      NASNU429    618
      L=LEN/L                                                          NASNU430    619
  805 FORMAT(37H1LABEL  COMP MDIST  DEGR  CONNECTIONS ,10X,            NASNU431    620
     +       20H(RENUMBERED NUMBERS) )                                 NASNU432    621
  810 FORMAT(5I6,20I5/ 25(25X,21I5/))                                  NASNU433    622
      DO 850 I=1,NN                                                    NASNU434    623
      IF(MOD(I,L).EQ.1) WRITE(6,805)                                   NASNU435    624
      DO 820 J=1,MAXD                                                  NASNU436    625
  820 IP(J)=0                                                          NASNU437    626
C CALCULATE MDIST AND PRINT TABLE.                                     NASNU438    627
      MDIST=0                                                          NASNU439    628
      DO 825 J=1,MAXD                                                  NASNU440    629
      K=IUNPK(IG,MAXGRD*(J-1)+I,NBITIN)                                NASNU441    630
      IF(K.EQ.0) GO TO 825                                             NASNU442    631
      MDIST=MAX0(MDIST,IABS(I-K))                                      NASNU443    632
      IP(J)=K                                                          NASNU444    633
  825 CONTINUE                                                         NASNU445    634
C INV(I,1)=IC(I)  BEFORE PACKING                                       NASNU446    635
C INV(MAXGRD+I,1)=IDEG(I)  BEFORE PACKING                              NASNU447    636
      IP1=INV(I,1)                                                     NASNU448    637
      IP2=INV(MAXGRD+I,1)                                              NASNU449    638
  850 WRITE(6,810) I,IP1,MDIST,IP2,(IP(J),J=1,MAXD)                    NASNU450    639
  900 RETURN                                                           NASNU451    640
      END                                                              NASNU452    641
      SUBROUTINE FLIP(LIST,N,INV,II3,ICC)                              FLIP   2    642
C CONVERT $-ARRAY LIST OF LENGTH N FROM ORIGINAL TO INTERNAL NUMBERS.  FLIP   3    643
      COMMON /A/ MAXGRD,MAXDEG,KMOD                                    FLIP   4    644
      DIMENSION LIST(1),INV(II3,2),ICC(1)                              FLIP   5    645
C CHECK FOR DUPLICATE AND ZERO ENTRIES AND REDUCE N IF NECESSARY.      FLIP   6    646
      CALL FIXIT(LIST,N)                                               FLIP   7    647
      IF(N.LE.0) RETURN                                                FLIP   8    648
      DO 20 I=1,N                                                      FLIP   9    649
      J=LIST(I)                                                        FLIP  10    650
      IF(J.LE.0) GO TO 30                                              FLIP  11    651
      LOC=J-1                                                          FLIP  12    652
   10 LOC=MOD(LOC,KMOD)+1                                              FLIP  13    653
      IF(INV(LOC,1).EQ.0) GO TO 30                                     FLIP  14    654
      IF(INV(LOC,1).NE.J) GO TO 10                                     FLIP  15    655
      K=INV(LOC,2)                                                     FLIP  16    656
      LIST(I)=ICC(K)                                                   FLIP  17    657
   20 CONTINUE                                                         FLIP  18    658
      RETURN                                                           FLIP  19    659
C ABORT BANDIT DUE TO ILLEGAL GRID POINT REFERENCE ON $-CONTROL CARD.   FLIP  20    660
   30 WRITE(6,40) J                                                    FLIP  21    661
   40 FORMAT(11H1GRID POINT ,I10,30H  APPEARING ON A $ CARD IS NOT  ,  FLIP  22    662
     +  25H A STRUCTURAL GRID POINT.  /13H FATAL ERROR.  )             FLIP  23    663
      CALL BOMBIT(8)                                                   FLIP  24    664
      END                                                              FLIP  25    665
      SUBROUTINE GOOGAN(KA,KB,NIN,NOUT)                                GOOGAN 2    666
C THIS ROUTINE READS A NASTRAN DATA DECK AND RIGHT-ADJUSTS ALL         GOOGAN 3    667
C      BULK DATA IN ITS FIELDS.                                        GOOGAN 4    668
C IN ADDITION, THE CALLING ARGUMENTS PROVIDE THE FOLLOWING OPTIONS -   GOOGAN 5    669
C      KA=1, PROCESS ALL CARDS IN THE NASTRAN DATA DECK, OR            GOOGAN 6    670
C        =2, PROCESS ONLY THOSE CARDS WITH A C OR G IN COLUMN 1,       GOOGAN 7    671
C              MPC CARDS, AND THOSE CONTINUATION CARDS WITH ALL        GOOGAN 8    672
C              NUMERIC FIELDS.  THE ENDDATA CARD IS WRITTEN IN ANY CASE.GOOGAN 9   673
C      KB = 1, CONVERT ALL 8 COLUMN FIELDS TO 16 COLUMN FIELDS, OR     GOOGAN10    674
C         = 2, THE FIELD WIDTHS REMAIN UNCHANGED.                      GOOGAN11    675
C      NIN = THE LOGICAL UNIT FROM WHICH THE INPUT DECK IS READ.       GOOGAN12    676
C      NOUT = THE LOGICAL UNIT ON WHICH THE OUTPUT IS WRITTEN.         GOOGAN13    677
C NEITHER NIN NOR NOUT ARE REWOUND IN THIS ROUTINE.                    GOOGAN14    678
C IF AN ASTERISK APPEARS IN FIELD 1 AFTER THE MNEMONIC, IT IS LEFT-    GOOGAN15    679
C      ADJUSTED AGAINST THE MNEMONIC.                                  GOOGAN16    680
C THE FOLLOWING TWO (2) CARDS ARE REQUIRED IN THE DATA DECK -          GOOGAN17    681
C      (1) A BEGIN BULK CARD TO INDICATE THE BEGINNING OF THE          GOOGAN18    682
C          BULK DATA DECK, AND                                         GOOGAN19    683
C      (2) AN ENDDATA CARD TO INDICATE THE END OF THE DATA DECK.       GOOGAN20    684
C ALL CARDS PRECEDING THE BEGIN BULK CARD ARE WRITTEN ON NOUT IFF KA=1. GOOGAN21    685
      DIMENSION ANUM(10)                                               GOOGAN22    686
      COMMON A(80),IP(40)                                              GOOGAN23    687
      COMMON IA,IB,ICARD,IFLAG,J,JNB,L,MKHOLD,MKINSR,MKNIN             GOOGAN24    688
      COMMON NBLANK,NFIELD,I,ICOL,IFIELD,IPROC,ITYPE                   GOOGAN25    689
      COMMON K,KAST,KBLK,MKI,MKJ,NCOL,NIP,NN                           GOOGAN26    690
      COMMON /A/ MAXGRD,MAXDEG,KMOD,NMPC                               GOOGAN27    691
      COMMON /B/ IPARAM(20),IARG(5)                                    GOOGAN28    692
      COMMON /DOL/ ISTART(100),IGNORE(100),IFIRST(100)                 GOOGAN29    693
      COMMON /DOLL/ IDIM,ISTA,IIG,IFIR,IGDEG,ISCH                      GOOGAN30    694
      COMMON /NG/ NGRID                                                GOOGAN31    695
      REAL M,N,II,LL,JJ,KK                                             GOOGAN32    696
      INTEGER EOF                                                      GOOGAN33    697
C DATA CARDS FOR ALPHABET (ALLOWS FOR FUTURE ADDITIONS TO              GOOGAN34    698
C            USER OPTION LIST).                                        GOOGAN35    699
      DATA B,C,D,E,G,M,N,P/1HB,1HC,1HD,1HE,1HG,1HM,1HN,1HP/            GOOGAN36    700
```

```
      DATA AA,II,LL,O,R/1HA,1HI,1HL,1HO,1HR/                            GOOGAN37   701
      DATA Q,S,T,U,Y/1HQ,1HS,1HT,1HU,1HY/                               GOOGAN38   702
      DATA F,H,JJ,KK/1HF,1HH,1HJ,1HK/                                   GOOGAN39   703
      DATA V,W,X,Z/1HV,1HW,1HX,1HZ/                                     GOOGAN40   704
      DATA ASTER,PLUS,BLANK,DOLLAR/1H*,1H+,1H ,1H$/                     GOOGAN41   705
      DATA ANUM/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/                GOOGAN42   706
      DATA LFLAG/0/                                                     GOOGAN43   707
      DATA IBOMB,IBOM/0,0/                                              GOOGAN44   708
      LFLAG=LFLAG+1                                                     GOOGAN45   709
    9 FORMAT(1H1)                                                       GOOGAN46   710
   10 FORMAT(80A1)                                                      GOOGAN47   711
   11 FORMAT(8A1,4(8X,8A1),3H*XZ,I5/3H*XZ,I5,4(8X,8A1),8A1)             GOOGAN48   712
      ICARD=0                                                           GOOGAN49   713
      MKINSR=12                                                         GOOGAN50   714
      MKNIN=NIN                                                         GOOGAN51   715
C READ EXECUTIVE OR CASE CONTROL CARD.                                  GOOGAN52   716
   20 READ(NIN,10)A                                                     GOOGAN53   717
      IF(EOF(NIN).EQ.0)GO TO 21                                         GOOGAN54   718
      IF(NIN.EQ.MKNIN)CALL BOMBIT(1)                                    GOOGAN55   719
      MKHOLD=NIN                                                        GOOGAN56   720
      NIN=MKINSR                                                        GOOGAN57   721
      MKINSR=MKHOLD                                                     GOOGAN58   722
      GO TO 20                                                          GOOGAN59   723
   21 IFLAG=0                                                           GOOGAN60   724
      ICARD=ICARD+1                                                     GOOGAN61   725
C PROCESS OUTPUT OPTION CARD, IF PRESENT.                               GOOGAN62   726
      IF(A(1).NE.DOLLAR)GO TO 26                                        GOOGAN63   727
      IF(LFLAG.GT.1) GO TO 29                                           GOOGAN64   728
C LOOK FOR FIRST KEYWORD.                                               GOOGAN65   729
      ITYPE=0                                                           GOOGAN66   730
      IF(A(2).EQ.P.AND.A(3).EQ.U)ITYPE=1                                GOOGAN67   731
      IF(A(2).EQ.M.AND.A(3).EQ.P)ITYPE=4                                GOOGAN68   732
      IF(A(2).EQ.S.AND.A(3).EQ.E)ITYPE=5                                GOOGAN69   733
      IF(A(2).EQ.R.AND.A(3).EQ.II)ITYPE=6                               GOOGAN70   734
      IF(A(2).EQ.N.AND.A(3).EQ.AA)ITYPE=8                               GOOGAN71   735
      IF(A(2).EQ.P.AND.A(3).EQ.R)ITYPE=10                               GOOGAN72   736
      IF(A(2).EQ.S.AND.A(3).EQ.C)GO TO 1100                             GOOGAN73   737
      IF(A(2).EQ.S.AND.A(3).EQ.T) GO TO 1200                            GOOGAN74   738
      IF(A(2).EQ.D.AND.A(3).EQ.E) GO TO 1250                            GOOGAN75   739
      IF(A(2).EQ.F.AND.A(3).EQ.II) GO TO 1300                           GOOGAN76   740
      IF(A(2).EQ.II.AND.A(3).EQ.G) GO TO 1350                           GOOGAN77   741
      IF(A(2).EQ.G.AND.A(3).EQ.R) GO TO 1380                            GOOGAN78   742
      IF(A(2).EQ.AA.AND.A(3).EQ.T) ITYPE=12                             GOOGAN79   743
      IF(A(2).EQ.W.AND.A(3).EQ.AA) ITYPE=13                             GOOGAN80   744
      IF(A(2).NE.II.OR.A(3).NE.N)GO TO 1025                             GOOGAN81   745
C     INSERT CARDS FROM ALTERNATE FILE                                  GOOGAN82   746
      IPARAM(11)=1                                                      GOOGAN83   747
      MKHOLD=NIN                                                        GOOGAN84   748
      NIN=MKINSR                                                        GOOGAN85   749
      MKINSR=MKHOLD                                                     GOOGAN86   750
      DO 1021 MKI=2,80                                                  GOOGAN87   751
      MKJ=81-MKI                                                        GOOGAN88   752
 1821 A(MKJ+1)=A(MKJ)                                                   GOOGAN89   753
      IPARAM(6)=4                                                       GOOGAN90   754
 1025 IF(ITYPE.EQ.0)GO TO 26                                            GOOGAN91   755
C LOOK FOR SECOND KEYWORD.                                              GOOGAN92   756
      I=3                                                               GOOGAN93   757
   22 I=I+1                                                             GOOGAN94   758
      IF(I.GE.79)GO TO 26                                               GOOGAN95   759
      IF(A(I).NE.BLANK)GO TO 22                                         GOOGAN96   760
   24 I=I+1                                                             GOOGAN97   761
      IF(I.GE.80)GO TO 26                                               GOOGAN98   762
      IF(A(I).EQ.BLANK)GO TO 24                                         GOOGAN99   763
      J=0                                                               GOOGA100   764
      IF(A(I).EQ.S.AND.A(I+1).EQ.E)J=1                                  GOOGA101   765
      IF(A(I).EQ.AA.AND.A(I+1).EQ.LL)J=2                                GOOGA102   766
      IF(A(I).EQ.N.AND.A(I+1).EQ.O)J=3                                  GOOGA103   767
      IF(A(I).EQ.Y.AND.A(I+1).EQ.E)J=4                                  GOOGA104   768
      IF(A(I).EQ.M.AND.A(I+1).EQ.II) J=5                                GOOGA105   769
      IF(A(I).EQ.M.AND.A(I+1).EQ.AA) J=6                                GOOGA106   770
      IF(J.EQ.0)GO TO 26                                                GOOGA107   771
C SET PARAMETER.                                                        GOOGA108   772
      IPARAM(ITYPE)=J                                                   GOOGA109   773
      GO TO 26                                                          GOOGA110   774
C READ $SCHEME CARD.                                                    GOOGA111   775
 1100 CALL READIT(A,IP,NIP)                                             GOOGA112   776
      ISCH=1                                                            GOOGA113   777
      I=MIN0(NIP,5)                                                     GOOGA114   778
      IF(I.EQ.0) GO TO 29                                               GOOGA115   779
      DO 1110 J=1,I                                                     GOOGA116   780
 1110 IARG(J)=IP(J)                                                     GOOGA117   781
      GO TO 29                                                          GOOGA118   782
C READ $START CARD.                                                     GOOGA119   783
 1200 CALL READIT(A,IP,NIP)                                             GOOGA120   784
      I=ISTA                                                            GOOGA121   785
      ISTA=ISTA+NIP                                                     GOOGA122   786
      IF(ISTA.LE.IDIM) GO TO 1205                                       GOOGA123   787
      IBOM=2                                                            GOOGA124   788
      ISTA=IDIM                                                         GOOGA125   789
      GO TO 29                                                          GOOGA126   790
 1205 DO 1210 J=1,NIP                                                   GOOGA127   791
 1210 ISTART(I+J)=IP(J)                                                 GOOGA128   792
      GO TO 29                                                          GOOGA129   793
C READ $DEGREE CARD.                                                    GOOGA130   794
 1250 CALL READIT(A,IP,NIP)                                             GOOGA131   795
      IGDEG=IP(1)                                                       GOOGA132   796
      GO TO 29                                                          GOOGA133   797
C READ $FIRST CARD.                                                     GOOGA134   798
 1300 CALL READIT(A,IP,NIP)                                             GOOGA135   799
      I=IFIR                                                            GOOGA136   800
```

```
      IFIR=IFIR+NIP                                                    GOOGA137   801
      IF(IFIR.LE.IDIM) GO TO 1308                                      GOOGA138   802
      IBOM=2                                                           GOOGA139   803
      IFIR=IDIM                                                        GOOGA140   804
      GO TO 29                                                         GOOGA141   805
 1308 DO 1310 J=1,NIP                                                  GOOGA142   806
 1310 IFIRST(I+J)=IP(J)                                                GOOGA143   807
      GO TO 29                                                         GOOGA144   808
C READ $IGNORE CARD.                                                   GOOGA145   809
 1350 CALL READIT(A,IP,NIP)                                            GOOGA146   810
      I=IIG                                                            GOOGA147   811
      IIG=IIG+NIP                                                      GOOGA148   812
      IF(IIG.LE.IDIM) GO TO 1360                                       GOOGA149   813
      IBOM=2                                                           GOOGA150   814
      IIG=IDIM                                                         GOOGA151   815
      GO TO 29                                                         GOOGA152   816
 1360 DO 1365 J=1,NIP                                                  GOOGA153   817
 1365 IGNORE(I+J)=IP(J)                                                GOOGA154   818
      GO TO 29                                                         GOOGA155   819
C READ $GRID CARD.                                                     GOOGA156   820
 1380 CALL READIT(A,IP,NIP)                                            GOOGA157   821
      NGRID=IP(1)                                                      GOOGA158   822
      GO TO 29                                                         GOOGA159   823
C LOOK FOR BEGIN BULK CARD.                                            GOOGA160   824
   26 I=0                                                              GOOGA161   825
   27 I=I+1                                                            GOOGA162   826
      IF(I.GT.75)GO TO 29                                              GOOGA163   827
      IF(A(I).EQ.BLANK)GO TO 27                                        GOOGA164   828
      IF(A(I).NE.B)GO TO 29                                            GOOGA165   829
      IF(A(I+1).NE.E)GO TO 29                                          GOOGA166   830
      IF(A(I+2).NE.G)GO TO 29                                          GOOGA167   831
      IF(A(I+3).NE.II)GO TO 29                                         GOOGA168   832
      IFLAG=1                                                          GOOGA169   833
C LEFT-ADJUST BEGIN BULK CARD.                                         GOOGA170   834
      K=73-I                                                           GOOGA171   835
      DO 28 J=1,72                                                     GOOGA172   836
      IF(J.LE.K)A(J)=A(J+I-1)                                          GOOGA173   837
   28 IF(J.GT.K)A(J)=BLANK                                             GOOGA174   838
      IF(LFLAG.GT.1) GO TO 29                                          GOOGA175   839
C REJECT ILLEGAL PARAMETERS AND SET TO DEFAULTS.                       GOOGA176   840
      IF(IPARAM(1).NE.2.AND.IPARAM(1).NE.3) IPARAM(1)=1                GOOGA177   841
      DO 1450 I=2,9                                                    GOOGA178   842
 1450 IF(IPARAM(I).NE.4) IPARAM(I)=3                                   GOOGA179   843
      IF(IPARAM(10).NE.6) IPARAM(10)=5                                 GOOGA180   844
      IF(IPARAM(12).NE.3) IPARAM(12)=4                                 GOOGA181   845
      IF(IPARAM(13).NE.4) IPARAM(13)=3                                 GOOGA182   846
      CALL GRID(NGRID)                                                 GOOGA183   847
      I=ISTA+IIG+IFIR+ISCH+IGDEG                                       GOOGA184   848
      IF(I.LE.0) GO TO 29                                              GOOGA185   849
C CHECK FOR ILLEGAL SCHEME ARGUMENTS.                                  GOOGA186   850
      DO 1460 I=1,3                                                    GOOGA187   851
 1460 IF(IARG(I).LT.1.OR.IARG(I).GT.MAXGRD) IBOMB=1                    GOOGA188   852
      IF(IARG(4).LT.2.OR.IARG(4).GT.3) IBOMB=1                         GOOGA189   853
      IF(IARG(5).LT.0.OR.IARG(5).GT.1) IBOMB=1                         GOOGA190   854
      WRITE(6,9)                                                       GOOGA191   855
      IF(ISCH.GT.0) WRITE(6,1500) (IARG(I),I=1,5)                      GOOGA192   856
 1500 FORMAT(//,9H $SCHEME ,10I10/200(9X,10I10/))                      GOOGA193   857
      IF(ISTA.GT.0) WRITE(6,1505) (ISTART(I),I=1,ISTA)                 GOOGA194   858
 1505 FORMAT(//,9H $START  ,10I10/200(9X,10I10/))                      GOOGA195   859
      IF(IGDEG.GT.0) WRITE(6,1510) IGDEG                               GOOGA196   860
 1510 FORMAT(//,9H $DEGREE ,10I10/200(9X,10I10/))                      GOOGA197   861
      IF(IFIR.GT.0) WRITE(6,1515) (IFIRST(I),I=1,IFIR)                 GOOGA198   862
 1515 FORMAT(//,9H $FIRST  ,10I10/200(9X,10I10/))                      GOOGA199   863
      IF(IIG.GT.0) WRITE(6,1520) (IGNORE(I),I=1,IIG)                   GOOGA200   864
 1520 FORMAT(//,9H $IGNORE ,10I10/200(9X,10I10/))                      GOOGA201   865
      IF(IBOMB.EQ.1) CALL BOMBIT(4)                                    GOOGA202   866
      IF(IBOM.EQ.2) CALL BOMBIT(9)                                     GOOGA203   867
   29 IF(KA.EQ.1)WRITE(NOUT,10)A                                       GOOGA204   868
      IF(IFLAG.EQ.0)GO TO 20                                           GOOGA205   869
C RETURN IF RIGHT-ADJUSTING OF CARDS IS NOT NEEDED.                    GOOGA206   870
      IF(IPARAM(5).EQ.3.AND.IPARAM(6).EQ.3)RETURN                      GOOGA207   871
C READ BULK DATA CARD.                                                 GOOGA208   872
   30 READ(NIN,10)A                                                    GOOGA209   873
      IF(EOF(NIN).EQ.0)GO TO 31                                        GOOGA210   874
      IF(NIN.EQ.MKNIN) CALL BOMBIT(1)                                  GOOGA211   875
C     SWITCH INPUT FILES                                               GOOGA212   876
      MKHOLD=NIN                                                       GOOGA213   877
      NIN=MKINSR                                                       GOOGA214   878
      MKINSR=MKHOLD                                                    GOOGA215   879
      GO TO 30                                                         GOOGA216   880
   31 ICARD=ICARD+1                                                    GOOGA217   881
C LEFT-ADJUST FIRST FIELD.                                             GOOGA218   882
      DO 1600 I=1,8                                                    GOOGA219   883
      IF(A(I).NE.BLANK)  GO TO 1610                                    GOOGA220   884
 1600 CONTINUE                                                         GOOGA221   885
      GO TO 30                                                         GOOGA222   886
 1610 IF(I.EQ.1) GO TO 1650                                            GOOGA223   887
      J=I-1                                                            GOOGA224   888
      K=8-J                                                            GOOGA225   889
      DO 1620 I=1,K                                                    GOOGA226   890
      A(I)=A(I+J)                                                      GOOGA227   891
 1620 A(I+J)=BLANK                                                     GOOGA228   892
 1650 CONTINUE                                                         GOOGA229   893
C LOOK FOR SEQGP CARD.                                                 GOOGA230   894
      IF(A(1).EQ.S.AND.A(2).EQ.E.AND.A(3).EQ.Q.AND.A(4).EQ.G)IPARAM(7)=4GOOGA231   895
C LOOK FOR COMMENT CARD.                                               GOOGA232   896
      IF(A(1).EQ.DOLLAR.AND.KA.EQ.1)GO TO 35                           GOOGA233   897
C LOOK FOR ENDDATA CARD.                                               GOOGA234   898
      I=0                                                              GOOGA235   899
   32 I=I+1                                                            GOOGA236   900
```

```
      IF(I.GT.75)GO TO 35                                              GOOGA237   901
      IF(A(I).EQ.BLANK)GO TO 32                                        GOOGA238   902
      IF(A(I).NE.E)GO TO 40                                            GOOGA239   903
      IF(A(I+1).NE.N)GO TO 40                                          GOOGA240   904
      IF(A(I+2).NE.D)GO TO 40                                          GOOGA241   905
      IF(A(I+3).NE.D)GO TO 40                                          GOOGA242   906
C LEFT-ADJUST ENDDATA CARD.                                            GOOGA243   907
      K=73-I                                                           GOOGA244   908
      DO 33 J=1,72                                                     GOOGA245   909
      IF(J.LE.K)A(J)=A(J+I-1)                                          GOOGA246   910
   33 IF(J.GT.K)A(J)=BLANK                                             GOOGA247   911
      WRITE(NOUT,10)A                                                  GOOGA248   912
      RETURN                                                           GOOGA249   913
   35 WRITE(NOUT,10)A                                                  GOOGA250   914
      GO TO 30                                                         GOOGA251   915
C DETERMINE IF CARD IS TO BE PROCESSED.                                GOOGA252   916
   40 IF(KA.EQ.1)GO TO 150                                             GOOGA253   917
      IF(A(1).EQ.C.OR.A(1).EQ.G)GO TO 150                              GOOGA254   918
      IF(A(1).EQ.M.AND.A(2).EQ.P)GO TO 150                             GOOGA255   919
      NCOL=8                                                           GOOGA256   920
      IF(A(1).EQ.ASTER)GO TO 50                                        GOOGA257   921
      IF(A(1).EQ.PLUS)GO TO 60                                         GOOGA258   922
      GO TO 30                                                         GOOGA259   923
   50 NCOL=16                                                          GOOGA260   924
   60 NFIELD=64/NCOL                                                   GOOGA261   925
      I=0                                                              GOOGA262   926
   70 I=I+1                                                            GOOGA263   927
      IF(I.GT.NFIELD)GO TO 150                                         GOOGA264   928
      IPROC=0                                                          GOOGA265   929
      IFLAG=0                                                          GOOGA266   930
      J=0                                                              GOOGA267   931
   80 J=J+1                                                            GOOGA268   932
      IF(IPROC.EQ.1)GO TO 70                                           GOOGA269   933
      IF(J.LE.NCOL)GO TO 90                                            GOOGA270   934
      IF(IFLAG.EQ.1)GO TO 30                                           GOOGA271   935
      GO TO 70                                                         GOOGA272   936
   90 ICOL=8+NCOL*(I-1)+J                                              GOOGA273   937
      IF(A(ICOL).EQ.BLANK)GO TO 80                                     GOOGA274   938
      IFLAG=1                                                          GOOGA275   939
      DO 100 L=1,10                                                    GOOGA276   940
  100 IF(A(ICOL).EQ.ANUM(L))IPROC=1                                    GOOGA277   941
      GO TO 80                                                         GOOGA278   942
C PROCESS FIRST FIELD.                                                 GOOGA279   943
  150 NCOL=8                                                           GOOGA280   944
      KAST=8                                                           GOOGA281   945
      KBLK=8                                                           GOOGA282   946
      DO 160 I=1,8                                                     GOOGA283   947
      IF(A(I).NE.BLANK.AND.A(I).NE.ASTER.AND.A(I+1).EQ.BLANK)KBLK=I+1  GOOGA284   948
      IF(A(I).EQ.ASTER)KAST=I                                          GOOGA285   949
  160 IF(A(I).EQ.ASTER)NCOL=16                                         GOOGA286   950
      IF(A(1).EQ.PLUS)NCOL=8                                           GOOGA287   951
      IF(NCOL.EQ.16)GO TO 170                                          GOOGA288   952
      IF(KB.EQ.2)GO TO 200                                             GOOGA289   953
      IF(A(1).NE.PLUS)A(KBLK)=ASTER                                    GOOGA290   954
      IF(A(1).EQ.PLUS)A(1)=ASTER                                       GOOGA291   955
      GO TO 200                                                        GOOGA292   956
  170 IF(A(1).EQ.ASTER)GO TO 200                                       GOOGA293   957
      IA=MIN0(KAST,KBLK)                                               GOOGA294   958
      IB=MAX0(KAST,KBLK)                                               GOOGA295   959
      A(IB)=BLANK                                                      GOOGA296   960
      A(IA)=ASTER                                                      GOOGA297   961
C RIGHT-ADJUST ALL BULK DATA WHICH IS TO BE PROCESSED.                 GOOGA298   962
  200 NFIELD=64/NCOL                                                   GOOGA299   963
      IFIELD=0                                                         GOOGA300   964
  210 IFIELD=IFIELD+1                                                  GOOGA301   965
      IF(IFIELD.GT.NFIELD)GO TO 300                                    GOOGA302   966
      I=0                                                              GOOGA303   967
  220 I=I+1                                                            GOOGA304   968
      IF(I.GT.NCOL)GO TO 210                                           GOOGA305   969
      ICOL=9+NCOL*IFIELD-I                                             GOOGA306   970
      IF(A(ICOL).EQ.BLANK)GO TO 220                                    GOOGA307   971
      NBLANK=I-1                                                       GOOGA308   972
      NN=NCOL-NBLANK                                                   GOOGA309   973
      DO 230 I=1,NCOL                                                  GOOGA310   974
      J=9+NCOL*IFIELD-I                                                GOOGA311   975
      JNB=J-NBLANK                                                     GOOGA312   976
      IF(I.LE.NN)A(J)=A(JNB)                                           GOOGA313   977
      IF(I.GT.NN)A(J)=BLANK                                            GOOGA314   978
  230 CONTINUE                                                         GOOGA315   979
      GO TO 210                                                        GOOGA316   980
C WRITE NEW CARD.                                                      GOOGA317   981
  300 IF(KB.EQ.1) A(73)=ASTER                                          GOOGA318   982
      IF(NCOL.EQ.8.AND.KB.EQ.1)GO TO 310                               GOOGA319   983
      WRITE(NOUT,10)A                                                  GOOGA320   984
      GO TO 30                                                         GOOGA321   985
  310 WRITE(NOUT,11)(A(I),I=1,40),ICARD,ICARD,(A(I),I=41,80)           GOOGA322   986
      GO TO 30                                                         GOOGA323   987
      END                                                              GOOGA324   988
      SUBROUTINE GRID(NGRID)                                           GRID   2   989
C PARTITION EXPANDABLE CORE.                                           GRID   3   990
      COMMON /BITS/ NBITIN,NBITEX                                      GRID   4   991
      COMMON /A/ MAXGRD,MAXDEG                                         GRID   5   992
      COMMON /K/ II(7),KOR                                             GRID   6   993
      MAX=16384                                                        GRID   7   994
      N=NGRID                                                          GRID   8   995
      IF(N.GT.2045) NBITIN=15                                          GRID   9   996
      IF(N.LE.510) NBITIN=10                                           GRID  10   997
      IF(N.LT.100) N=100                                               GRID  11   998
      IF(N.GT.MAX) GO TO 40                                            GRID  12   999
C CALCULATE WIDTH II(2) OF IG MATRIX.                                  GRID  13  1000
```

```
 20     L=60/NBITIN                                                      GRID  14        1001
        M=60/NBITEX                                                      GRID  15        1002
        N=N+L*M-1                                                        GRID  16        1003
        N=N-MOD(N,L*M)                                                   GRID  17        1004
        MAXGRO=N                                                         GRID  18        1005
C  I=PACKED LENGTH FOR INTERNAL NUMBER.                                  GRID  19        1006
C  J=PACKED LENGTH FOR ORIGINAL NUMBER.                                  GRID  20        1007
        I=N/L                                                            GRID  21        1008
        J=N/M                                                            GRID  22        1009
C SET UP DIMENSIONS IN II ARRAY, WHERE IG(II1,II2),INV(II3,2),           GRID  23        1010
C    INT(II4),ICC(II5),ILD(II6),NORIG(II7)                               GRID  24        1011
        II(1)=I                                                          GRID  25        1012
        II(3)=2*J                                                        GRID  26        1013
        II(4)=J                                                          GRID  27        1014
        II(5)=J                                                          GRID  28        1015
        II(6)=J                                                          GRID  29        1016
        II(7)=J                                                          GRID  30        1017
        I=2*II(3)+II(4)+II(5)+II(6)+II(7)                                GRID  31        1018
        II(2)=(KOR-I)/(II(1)+2)                                          GRID  32        1019
C DENOMINATOR CONTAINS A 2 TO ALLOW FOR 2 SCRATCH ARRAYS, EACH OF        GRID  33        1020
C           LENGTH MAXDEG.                                               GRID  34        1021
        II(2)=MIN0(II(2),N-1)                                            GRID  35        1022
        MAXDEG=II(2)                                                     GRID  36        1023
        RETURN                                                           GRID  37        1024
C SUBSTITUTE MAX IF NGRID TOO LARGE.                                     GRID  38        1025
 40     N=MAX                                                            GRID  39        1026
        WRITE(6,50) NGRID,N                                              GRID  40        1027
    50 FORMAT(23H1BANDIT WARNING MESSAGE/10X,6H$GRID ,I10,5X,            GRID  41        1028
       +      9HTOO LARGE /10X,6H$GRID ,I10,5X,12HSUBSTITUTED. )         GRID  42        1029
        GO TO 20                                                         GRID  43        1030
        END                                                              GRID  44        1031
        SUBROUTINE READIT(A,IP,NIP)                                      READIT 2        1032
C THIS ROUTINE READS AND STORES (IN IP) NUMERIC DATA APPEARING ON        READIT 3        1033
C    $-CONTROL CARDS UP TO COLUMN 72.                                    READIT 4        1034
        DIMENSION ANUM(10)                                               READIT 5        1035
        DIMENSION A(1),IP(1)                                             READIT 6        1036
        DATA ANUM/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/               READIT 7        1037
C INITIALIZE ARRAY.                                                      READIT 8        1038
        NIP=0                                                            READIT 9        1039
        DO 10 I=1,40                                                     READIT10        1040
    10 IP(I)=0                                                           READIT11        1041
        I=3                                                              READIT12        1042
        DO 70 KOUNT=1,40                                                 READIT13        1043
        NUM=0                                                            READIT14        1044
        NUMFL=0                                                          READIT15        1045
    20 I=I+1                                                             READIT16        1046
        IF(I.LE.72) GO TO 30                                             READIT17        1047
        IF(NUMFL.EQ.1) GO TO 60                                          READIT18        1048
        RETURN                                                           READIT19        1049
    30 K=99                                                              READIT20        1050
        DO 40 J=1,10                                                     READIT21        1051
    40 IF(A(I).EQ.ANUM(J)) K=J-1                                         READIT22        1052
        IF(K.NE.99) GO TO 50                                             READIT23        1053
        IF(NUMFL) 60,20,60                                               READIT24        1054
    50 NUMFL=1                                                           READIT25        1055
        NUM=10*NUM+K                                                     READIT26        1056
        GO TO 20                                                         READIT27        1057
    60 NIP=KOUNT                                                         READIT28        1058
        IP(NIP)=NUM                                                      READIT29        1059
    70 CONTINUE                                                          READIT30        1060
        NIP=40                                                           READIT31        1061
        RETURN                                                           READIT32        1062
        END                                                              READIT33        1063
        SUBROUTINE BOMBIT(IERR)                                          BOMBIT 2        1064
C BOMB BANDIT TO SUPPRESS THE EXECUTION OF NASTRAN.                      BOMBIT 3        1065
        COMMON /B/ IPARAM(20)                                            BOMBIT 4        1066
        COMMON /K/ II(7),KORE,IFL                                        BOMBIT 5        1067
     3 FORMAT(*-CURRENT FIELD LENGTH (FL) =  *,06,*B*/                   BOMBIT 6        1068
       + * THIS BANDIT JOB MAY REQUIRE A LARGER FIELD LENGTH (FL)*/      BOMBIT 7        1069
       + *-THEREFORE, MAKE THE FOLLOWING CHANGES*/                       BOMBIT 8        1070
       + *       1.  INCREASE THE FL*/                                   BOMBIT 9        1071
       + *       2.  INSERT A NOREDUCE. CARD IMMEDIATELY BEFORE THE *,   BOMBIT10        1072
       + *BANDIT. CARD*/                                                 BOMBIT11        1073
       + *       3.  INSERT A $GRID N CARD SOMEWHERE BEFORE THE BEGIN*,  BOMBIT12        1074
       + * BULK CARD, WHERE THE INTEGER N IS AN UPPER BOUND (PREFERABLY*/ BOMBIT13        1075
       + *           LEAST UPPER BOUND) ON THE NUMBER OF GRID POINTS*)   BOMBIT14        1076
     5 FORMAT(200(1H+,130X/))                                            BOMBIT15        1077
        CALL REMARK(40H   ************************************ )        BOMBIT16        1078
        GO TO (10,20,30,40,50,60,70,80,90), IERR                         BOMBIT17        1079
  EOF ENCOUNTERED.                                                       BOMBIT18        1080
    10 WRITE(6,12)                                                       BOMBIT19        1081
    12 FORMAT(55H1BANDIT FATAL ERROR -     MISSING BEGIN BULK OR ENDDATA,BOMBIT20        1082
       +    6H CARD. )                                                   BOMBIT21        1083
        CALL REMARK(39H   **MISSING BEGIN BULK OR ENDDATA CARD )         BOMBIT22        1084
        GO TO 500                                                        BOMBIT23        1085
C BULK DATA CARD OUT OF SORT.                                            BOMBIT24        1086
    20 CALL REMARK(31H   **BULK DATA CARD OUT OF SORT )                  BOMBIT25        1087
        GO TO 500                                                        BOMBIT26        1088
C SEQGP CARDS IN DECK AND RESEQUENCING REQUESTED.                        BOMBIT27        1089
    30 CALL REMARK(32H   **SEQGP CARDS ALREADY IN DECK )                 BOMBIT28        1090
        GO TO 500                                                        BOMBIT29        1091
C $SCHEME ILLEGAL ARGUMENTS.                                             BOMBIT30        1092
    40 WRITE(6,42)                                                       BOMBIT31        1093
    42 FORMAT(46H1BANDIT FATAL ERROR -     ILLEGAL ARGUMENTS ON,         BOMBIT32        1094
       +   14H $SCHEME CARD. )                                           BOMBIT33        1095
        CALL REMARK(30H   **ILLEGAL $SCHEME ARGUMENTS )                  BOMBIT34        1096
        GO TO 500                                                        BOMBIT35        1097
C  TOO MANY TERMS IN MPC EQUATION.                                       BOMBIT36        1098
    50 CALL REMARK(36H   **MPC EQUATION HAS TOO MANY TERMS )             BOMBIT37        1099
        GO TO 500                                                        BOMBIT38        1100
```

```
C MAXDEG EXCEEDED.                                                    BOMBIT39    1101
   60 CALL REMARK(28H    **MAXIMUM DEGREE EXCEEDED )                  BOMBIT40    1102
      WRITE(6,3) IFL                                                  BOMBIT41    1103
      GO TO 500                                                       BOMBIT42    1104
C MAXGRD EXCEEDED.                                                    BOMBIT43    1105
   70 CALL REMARK(39H    **MAX NUMBER OF GRID POINTS EXCEEDED )       BOMBIT44    1106
      WRITE(6,3) IFL                                                  BOMBIT45    1107
      GO TO 500                                                       BOMBIT46    1108
C NON-EXISTENT GRID POINT REFERENCE ON $-CARD                         BOMBIT47    1109
   80 CALL REMARK(32H    **ILLEGAL REFERENCE ON $-CARD )              BOMBIT48    1110
      GO TO 500                                                       BOMBIT49    1111
C TOO MANY GRID POINTS ON $-CARD.                                     BOMBIT50    1112
   90 WRITE(6,92)                                                     BOMBIT51    1113
   92 FORMAT(51H1BANDIT FATAL ERROR -     TOO MANY POINTS ON $-CARD)  BOMBIT52    1114
      CALL REMARK(30H    **TOO MANY POINTS ON $-CARD )                BOMBIT53    1115
      GO TO 500                                                       BOMBIT54    1116
C ABORT BANDIT.                                                       BOMBIT55    1117
  500 CALL REMARK(17H    **BANDIT ABORT )                             BOMBIT56    1118
      CALL REMARK(23H    **NASTRAN SUPPRESSED )                       BOMBIT57    1119
      CALL REMARK(40H    ************************************ )       BOMBIT58    1120
      WRITE(6,5)                                                      BOMBIT59    1121
      CALL ABT                                                        BOMBIT65    1122
      STOP                                                            BOMBIT66    1123
      END                                                             BOMBIT67    1124
      SUBROUTINE SCAT(KG,NCON,NEW,INV,II3,NORIG)                      SCAT   2    1125
C THIS ROUTINE USES SCATTER SORT TECHNIQUES FOR EACH GRID POINT       SCAT   3    1126
C          ENCOUNTERED TO DETERMINE WHETHER OR NOT THE POINT HAS      SCAT   4    1127
C          BEEN SEEN BEFORE.  IF NOT, INV, NORIG, AND NEW ARE UPDATED.SCAT   5    1128
C INV(I,1) CONTAINS AN ORIGINAL GRID POINT NUMBER                     SCAT   6    1129
C INV(I,2) CONTAINS THE INTERNAL NUMBER ASSIGNED TO IT (BEFORE SORTING)SCAT   7    1130
      DIMENSION INV(II3,2),NORIG(1)                                   SCAT   8    1131
      COMMON /A/ MAXGRD,MAXDEG,KMOD                                   SCAT   9    1132
      DIMENSION KG(1)                                                 SCAT  10    1133
      DO 100 I=1,NCON                                                 SCAT  11    1134
      NOLD=KG(I)                                                      SCAT  12    1135
      IF(NOLD.EQ.0)GO TO 100                                          SCAT  13    1136
      LOC=NOLD-1                                                      SCAT  14    1137
   10 LOC=MOD(LOC,KMOD)+1                                             SCAT  15    1138
20    IF(INV(LOC,1).NE.0) GO TO 30                                    SCAT  16    1139
      INV(LOC,1)=NOLD                                                 SCAT  17    1140
      NEW=NEW+1                                                       SCAT  18    1141
      IF(NEW.GT.MAXGRD) GO TO 150                                     SCAT  19    1142
      NORIG(NEW)=NOLD                                                 SCAT  20    1143
      INV(LOC,2)=NEW                                                  SCAT  21    1144
      GO TO 40                                                        SCAT  22    1145
30    IF(INV(LOC,1).NE.NOLD) GO TO 10                                 SCAT  23    1146
40    KG(I)=INV(LOC,2)                                                SCAT  24    1147
  100 CONTINUE                                                        SCAT  25    1148
      RETURN                                                          SCAT  26    1149
  150 WRITE(6,160) MAXGRD                                             SCAT  27    1150
  160 FORMAT(35H1 THIS STRUCTURE CONTAINS MORE THAN,I6,               SCAT  28    1151
     +  14H  GRID POINTS. /14H  FATAL ERROR. )                        SCAT  29    1152
      CALL BOMBIT(7)                                                  SCAT  30    1153
      END                                                             SCAT  31    1154
      SUBROUTINE BRIGIT(IG,II1,INV,II3,INT,ICC,NORIG,IP)              BRIGIT 2    1155
C THIS ROUTINE GENERATES A NEW INTERNAL/EXTERNAL CORRESPONDENCE       BRIGIT 3    1156
C      TABLE NORIG AND CONNECTION TABLE IG SUCH THAT THE NEW INTERNAL BRIGIT 4    1157
C      NUMBERS CORRESPOND TO A SORT OF THE ORIGINAL NUMBERS INTO      BRIGIT 5    1158
C      ASCENDING ORDER.                                               BRIGIT 6    1159
C INPUT - IG,INV,NORIG                                                BRIGIT 7    1160
C OUTPUT - IG,NORIG,ICC                                               BRIGIT 8    1161
C SCRATCH - INT,IP                                                    BRIGIT 9    1162
      DIMENSION IG(II1,1),INV(II3,2)                                  BRIGIT10    1163
      DIMENSION INT(1),ICC(1),NORIG(1),IP(1)                          BRIGIT11    1164
      COMMON /S/ NN,MM,IH,IB                                          BRIGIT12    1165
      COMMON /A/ MAXGRD,MAXDEG,KMOD,NMPC                              BRIGIT13    1166
      COMMON /BITS/ NBITIN,NBITEX,IPASS                               BRIGIT14    1167
      REWIND 8                                                        BRIGIT15    1168
C PERFORM A ROUGH SORT OF THE ORIGINAL GRID NUMBERS.                  BRIGIT16    1169
      L=0                                                             BRIGIT17    1170
      KFAC=-1                                                         BRIGIT18    1171
   20 KFAC=KFAC+1                                                     BRIGIT19    1172
      MIN=2147483647                                                  BRIGIT20    1173
      DO 50 I=1,KMOD                                                  BRIGIT21    1174
      IF(INV(I,1).GT.(KFAC*KMOD))                                     BRIGIT22    1175
     +      MIN=MIN0(MIN,INV(I,1))                                    BRIGIT23    1176
   50 CONTINUE                                                        BRIGIT24    1177
      KFAC=(MIN-1)/KMOD                                               BRIGIT25    1178
      DO 80 I=1,KMOD                                                  BRIGIT26    1179
      IS=INV(I,1)                                                     BRIGIT27    1180
      IF(IS.LE.(KFAC*KMOD).OR.IS.GT.(KFAC+1)*KMOD)GO TO 80            BRIGIT28    1181
      L=L+1                                                           BRIGIT29    1182
      INT(L)=INV(I,1)                                                 BRIGIT30    1183
   80 CONTINUE                                                        BRIGIT31    1184
      IF(L.LT.NN)GO TO 20                                             BRIGIT32    1185
C COMPLETE THE SORTING OF THE ORIGINAL GRID NUMBERS.                  BRIGIT33    1186
      CALL SORT(INT,NN)                                               BRIGIT34    1187
C DETERMINE CORRESPONDENCE (ICC) BETWEEN NORIG AND INT ARRAYS.        BRIGIT35    1188
      DO 130 I=1,NN                                                   BRIGIT36    1189
      L=INT(I)                                                        BRIGIT37    1190
      LOC=L-1                                                         BRIGIT38    1191
  110 LOC=MOD(LOC,KMOD)+1                                             BRIGIT39    1192
120   IF(INV(LOC,1).NE.L) GO TO 110                                   BRIGIT40    1193
      M=INV(LOC,2)                                                    BRIGIT41    1194
      ICC(M)=I                                                        BRIGIT42    1195
  130 CONTINUE                                                        BRIGIT43    1196
C TRANSFER INT ARRAY TO NORIG ARRAY.                                  BRIGIT44    1197
      DO 220 I=1,NN                                                   BRIGIT45    1198
  220 NORIG(I)=INT(I)                                                 BRIGIT46    1199
C CHANGE IG MATRIX ACCORDING TO CORRESPONDENCE TABLE ICC.             BRIGIT47    1200
```

```
      CALL SWITCH(IG,II1,INT,ICC,IP(1),IP(MAXDEG+1))                    BRIGIT48      1201
      REWIND 8                                                          BRIGIT49      1202
      RETURN                                                            BRIGIT50      1203
      END                                                               BRIGIT51      1204
      SUBROUTINE SORT(LIST,NL)                                          SORT   2      1205
C THIS SUBROUTINE SORTS A LIST OF LENGTH NL AND IS BIASED TOWARDS THOSE SORT   3      1206
C   LISTS NOT BADLY OUT OF SORT.                                        SORT   4      1207
      DIMENSION LIST(1)                                                 SORT   5      1208
      IF(NL.LE.1) RETURN                                                SORT   6      1209
      NL1=NL-1                                                          SORT   7      1210
      DO 20 I=1,NL1                                                     SORT   8      1211
      K=NL-I                                                            SORT   9      1212
      KFLAG=0                                                           SORT  10      1213
      DO 10 J=1,K                                                       SORT  11      1214
      IF(LIST(J).LE.LIST(J+1)) GO TO 10                                 SORT  12      1215
      KFLAG=1                                                           SORT  13      1216
      L=LIST(J)                                                         SORT  14      1217
      LIST(J)=LIST(J+1)                                                 SORT  15      1218
      LIST(J+1)=L                                                       SORT  16      1219
   10 CONTINUE                                                          SORT  17      1220
      IF(KFLAG.EQ.0) RETURN                                             SORT  18      1221
   20 CONTINUE                                                          SORT  19      1222
      RETURN                                                            SORT  20      1223
      END                                                               SORT  21      1224
      SUBROUTINE SETIG(KG1,KG2,IG,II1,NORIG)                            SETIG  2      1225
C THIS ROUTINE SETS IG(KG1,-)=KG2 AND IG(KG2,-)=KG1 IF THIS             SETIG  3      1226
C          CONNECTION HAS NOT ALREADY BEEN SET.                         SETIG  4      1227
      DIMENSION IG(II1,1),NORIG(1)                                      SETIG  5      1228
      COMMON /S/ NN,MM,IH,IB                                            SETIG  6      1229
      COMMON /A/ MAXGRD,MAXDEG,KMOD,NMPC                                SETIG  7      1230
      COMMON /BITS/ NBITIN,NBITEX,IPASS                                 SETIG  8      1231
      IF(KG1.EQ.0)RETURN                                                SETIG  9      1232
      IF(KG2.EQ.0)RETURN                                                SETIG 10      1233
      IF(KG1.EQ.KG2)RETURN                                              SETIG 11      1234
      DO 50 LOOP=1,2                                                    SETIG 12      1235
      L=KG1                                                             SETIG 13      1236
      K=KG2                                                             SETIG 14      1237
      IF(LOOP.EQ.1) GO TO 20                                            SETIG 15      1238
      L=KG2                                                             SETIG 16      1239
      K=KG1                                                             SETIG 17      1240
   20 M=0                                                               SETIG 18      1241
   30 M=M+1                                                             SETIG 19      1242
      IF(M.GT.MAXDEG) GO TO 60                                          SETIG 20      1243
      IS=IUNPK(IG,MAXGRD*(M-1)+L,NBITIN)                                SETIG 21      1244
      IF(IS.EQ.0) GO TO 40                                              SETIG 22      1245
      IF(IS.NE.K) GO TO 30                                              SETIG 23      1246
      GO TO 50                                                          SETIG 24      1247
   40 CALL PACK(IG,MAXGRD*(M-1)+L,NBITIN,K)                             SETIG 25      1248
      MM=MAX0(MM,M)                                                     SETIG 26      1249
   50 CONTINUE                                                          SETIG 27      1250
      RETURN                                                            SETIG 28      1251
   60 WRITE(6,70) NORIG(L),MAXDEG                                       SETIG 29      1252
   70 FORMAT(12H1 GRID POINT,I12,26H   HAS DEGREE GREATER THAN,I6/      SETIG 30      1253
     + 14H  FATAL ERROR. )                                              SETIG 31      1254
      CALL BOMBIT(6)                                                    SETIG 32      1255
      END                                                               SETIG 33      1256
      SUBROUTINE TIGER(NEQ,IG,II1,LIST,NORIG)                           TIGER  2      1257
C THIS ROUTINE MAKES ADDITIONS TO THE CONNECTION TABLE IG TO REFLECT     TIGER  3      1258
C     THE PRESENCE OF MPC'S AND STORES THE DEPENDENT POINTS IN LIST.     TIGER  4      1259
C NEQ=NUMBER OF MPC EQUATIONS.                                          TIGER  5      1260
      DIMENSION IG(II1,1),LIST(1),NORIG(1)                              TIGER  6      1261
      COMMON /S/ NN,MM,IH,IB                                            TIGER  7      1262
      COMMON /A/ MAXGRD,MAXDEG,KMOD,NMPC                                TIGER  8      1263
      COMMON /BITS/ NBITIN,NBITEX,IPASS                                 TIGER  9      1264
      DIMENSION KG(40)                                                  TIGER 10      1265
      IF(NEQ.EQ.0)RETURN                                                TIGER 11      1266
      REWIND 11                                                         TIGER 12      1267
C INITIALIZE LIST.                                                      TIGER 13      1268
      DO 20 I=1,NN                                                      TIGER 14      1269
   20 LIST(I)=0                                                         TIGER 15      1270
C GENERATE NEW CONNECTIONS.                                             TIGER 16      1271
      DO 100 II=1,NEQ                                                   TIGER 17      1272
      READ(11)KG                                                        TIGER 18      1273
      IGRID=KG(1)                                                       TIGER 19      1274
      LIST(IGRID)=IGRID                                                 TIGER 20      1275
      DO 100 I=1,MAXDEG                                                 TIGER 21      1276
      L=IUNPK(IG,MAXGRD*(I-1)+IGRID,NBITIN)                             TIGER 22      1277
      DO 100 J=2,NMPC                                                   TIGER 23      1278
  100 CALL SETIG(L,KG(J),IG,II1,NORIG)                                  TIGER 24      1279
      REWIND 11                                                         TIGER 25      1280
      RETURN                                                            TIGER 26      1281
      END                                                               TIGER 27      1282
      SUBROUTINE SWITCH(IG,II1,IFLAG,KT,KA,KB)                          SWITCH 2      1283
C THIS SUBROUTINE GENERATES A NEW IG MATRIX ACCORDING TO THE            SWITCH 3      1284
C      CORRESPONDENCE TABLE KT, WHICH MUST BE SET UP                    SWITCH 4      1285
C      PRIOR TO THE CALL.    ONLY INTERNAL NUMBERS ARE ALLOWED          SWITCH 5      1286
C      AS VALUES OF KT.                                                 SWITCH 6      1287
C                                                                       SWITCH 7      1288
C INPUT - IG,KT                                                         SWITCH 8      1289
C OUTPUT - IG                                                           SWITCH 9      1290
C SCRATCH - IFLAG,KA,KB                                                 SWITCH10      1291
C                                                                       SWITCH11      1292
      DIMENSION IG(II1,1),IFLAG(1),KT(1),KA(1),KB(1)                    SWITCH12      1293
      COMMON /S/ NN,MM,IH,IB                                            SWITCH13      1294
      COMMON /A/ MAXGRD,MAXDEG,KMOD,NMPC                                SWITCH14      1295
      COMMON /BITS/ NBITIN,NBITEX,IPASS                                 SWITCH15      1296
C KT=CORRESPONDENCE TABLE.  KT(OLD) = NEW.                              SWITCH16      1297
C KA,KB = TEMPORARY STORAGE ROWS.                                       SWITCH17      1298
      DO 100 I=1,NN                                                     SWITCH18      1299
      DO 90 J=1,MM                                                      SWITCH19      1300
```

```
      L=IUNPK(IG,MAXGRD*(J-1)+I,NBITIN)                                 SWITCH20  1301
      IF(L.LE.0) GO TO 100                                              SWITCH21  1302
      IS=KT(L)                                                          SWITCH22  1303
      CALL PACK(IG,MAXGRD*(J-1)+I,NBITIN,IS)                            SWITCH23  1304
   90 CONTINUE                                                          SWITCH24  1305
  100 CONTINUE                                                          SWITCH25  1306
C INITIALIZE FLAGS.                                                     SWITCH26  1307
      DO 120 I=1,NN                                                     SWITCH27  1308
  120 IFLAG(I)=0                                                        SWITCH28  1309
C INITIALIZE TEMPORARY STORAGE ROWS.                                    SWITCH29  1310
      DO 130 I=1,MM                                                     SWITCH30  1311
      KA(I)=0                                                           SWITCH31  1312
  130 KB(I)=0                                                           SWITCH32  1313
C RE-ORDER ROWS OF IG MATRIX.                                           SWITCH33  1314
      DO 200 IROW=1,NN                                                  SWITCH34  1315
      IF(IFLAG(IROW).EQ.1) GO TO 200                                    SWITCH35  1316
      IF(KT(IROW).EQ.IROW) GO TO 200                                    SWITCH36  1317
      IFLAG(IROW)=1                                                     SWITCH37  1318
      DO 140 J=1,MM                                                     SWITCH38  1319
140    KB(J)=IUNPK(IG,MAXGRD*(J-1)+IROW,NBITIN)                         SWITCH39  1320
      L=KT(IROW)                                                        SWITCH40  1321
150   IFLAG(L)=1                                                        SWITCH41  1322
      DO 160 J=1,MM                                                     SWITCH42  1323
      KA(J)=IUNPK(IG,MAXGRD*(J-1)+L,NBITIN)                             SWITCH43  1324
      CALL PACK(IG,MAXGRD*(J-1)+L,NBITIN,KB(J))                         SWITCH44  1325
  160 KB(J)=KA(J)                                                       SWITCH45  1326
      M=KT(L)                                                           SWITCH46  1327
      IF(IFLAG(M).EQ.1) GO TO 170                                       SWITCH47  1328
      L=M                                                               SWITCH48  1329
      GO TO 150                                                         SWITCH49  1330
  170 DO 180 J=1,MM                                                     SWITCH50  1331
  180 CALL PACK(IG,MAXGRD*(J-1)+M,NBITIN,KB(J))                         SWITCH51  1332
  200 CONTINUE                                                          SWITCH52  1333
      RETURN                                                            SWITCH53  1334
      END                                                               SWITCH54  1335
      SUBROUTINE MORRIS(LIST,NL,IG,II1)                                 MORRIS 2  1336
C THIS ROUTINE DELETES ALL REFERENCE IN THE CONNECTION TABLE IG         MORRIS 3  1337
C      TO THOSE POINTS IN A LIST OF LENGTH NL.                          MORRIS 4  1338
      DIMENSION IG(II1,1),LIST(1)                                       MORRIS 5  1339
      COMMON /S/ NN,MM                                                  MORRIS 6  1340
      COMMON /A/ MAXGRD                                                 MORRIS 7  1341
      COMMON /BITS/ NBITIN,NBITEX                                       MORRIS 8  1342
C COMPRESS OUT DUPLICATE ENTRIES IN LIST.                               MORRIS 9  1343
      CALL FIXIT(LIST,NL)                                               MORRIS10  1344
      IF(NL.LE.0) RETURN                                                MORRIS11  1345
      MM1=MM-1                                                          MORRIS12  1346
      DO 60 IJ=1,NL                                                     MORRIS13  1347
      I=LIST(IJ)                                                        MORRIS14  1348
      DO 50 J=1,MM                                                      MORRIS15  1349
      L=IUNPK(IG,MAXGRD*(J-1)+I,NBITIN)                                 MORRIS16  1350
      IF(L.EQ.0) GO TO 60                                               MORRIS17  1351
      K=0                                                               MORRIS18  1352
   20 K=K+1                                                             MORRIS19  1353
      M=IUNPK(IG,MAXGRD*(K-1)+L,NBITIN)                                 MORRIS20  1354
      IF(M.NE.I) GO TO 20                                               MORRIS21  1355
      IF(K.GE.MM) GO TO 40                                              MORRIS22  1356
      DO 30 N=K,MM1                                                     MORRIS23  1357
      IS=IUNPK(IG,MAXGRD*N+L,NBITIN)                                    MORRIS24  1358
   30 CALL PACK(IG,MAXGRD*(N-1)+L,NBITIN,IS)                            MORRIS25  1359
   40 CALL PACK(IG,MAXGRD*MM1+L,NBITIN,0)                               MORRIS26  1360
      CALL PACK(IG,MAXGRD*(J-1)+I,NBITIN,0)                             MORRIS27  1361
   50 CONTINUE                                                          MORRIS28  1362
   60 CONTINUE                                                          MORRIS29  1363
      RETURN                                                            MORRIS30  1364
      END                                                               MORRIS31  1365
      SUBROUTINE FIXIT(LIST,NL)                                         FIXIT  2  1366
C THIS ROUTINE COMPRESSES OUT ZEROES AND MULTIPLE ENTRIES IN A LIST     FIXIT  3  1367
C   ORIGINALLY OF LENGTH NL.    A CORRECTED LENGTH NL IS RETURNED TO    FIXIT  4  1368
C   THE CALLING PROGRAM.                                                FIXIT  5  1369
      DIMENSION LIST(1)                                                 FIXIT  6  1370
      IF(NL.LE.0) RETURN                                                FIXIT  7  1371
      IF(NL.EQ.1) GO TO 110                                             FIXIT  8  1372
      NL1=NL-1                                                          FIXIT  9  1373
C DELETE DUPLICATE ENTRIES.                                             FIXIT 10  1374
      DO 20 I=1,NL1                                                     FIXIT 11  1375
      IF(LIST(I).EQ.0) GO TO 20                                         FIXIT 12  1376
      I1=I+1                                                            FIXIT 13  1377
      DO 10 J=I1,NL                                                     FIXIT 14  1378
      IF(LIST(I).NE.LIST(J)) GO TO 10                                   FIXIT 15  1379
      LIST(I)=0                                                         FIXIT 16  1380
      GO TO 20                                                          FIXIT 17  1381
10    CONTINUE                                                          FIXIT 18  1382
20    CONTINUE                                                          FIXIT 19  1383
C DELETE ZEROES.                                                        FIXIT 20  1384
      DO 40 I=1,NL1                                                     FIXIT 21  1385
      K=0                                                               FIXIT 22  1386
25    IF(LIST(I).NE.0) GO TO 40                                         FIXIT 23  1387
      K=K+1                                                             FIXIT 24  1388
      DO 30 J=I,NL1                                                     FIXIT 25  1389
30    LIST(J)=LIST(J+1)                                                 FIXIT 26  1390
      LIST(NL)=0                                                        FIXIT 27  1391
      IF(K.GE.(NL-I+1)) GO TO 70                                        FIXIT 28  1392
      GO TO 25                                                          FIXIT 29  1393
40    CONTINUE                                                          FIXIT 30  1394
C CALCULATE NEW LENGTH NL.                                              FIXIT 31  1395
   70 DO 80 I=1,NL                                                      FIXIT 32  1396
      J=NL-I+1                                                          FIXIT 33  1397
      IF(LIST(J).NE.0) GO TO 90                                         FIXIT 34  1398
   80 CONTINUE                                                          FIXIT 35  1399
   90 NL=NL-I+1                                                         FIXIT 36  1400
```

```
      RETURN                                                            FIXIT 37        1401
  110 IF(LIST(1).EQ.0) NL=0                                             FIXIT 38        1402
      RETURN                                                            FIXIT 39        1403
      END                                                               FIXIT 40        1404
      SUBROUTINE SCHEME(NT,NUM,NOM,IO,IP,IG,II1,IC,IDEG,IDIS,IW,        SCHEME 2        1405
     + NEW,ICC,ILD,IPP)                                                 SCHEME 3        1406
C  IO IS VALID IFF 2.LE.IO.LE.3                                         SCHEME 4        1407
      DIMENSION IG(II1,1),IC(1),IDEG(1),IDIS(1),IW(1)                   SCHEME 5        1408
      DIMENSION NEW(1),ICC(1),ILD(1),IPP(1)                             SCHEME 6        1409
C IPP HAS DIMENSION 2*MAXDEG                                            SCHEME 7        1410
      COMMON /S/ NN,MM,IH,IB                                            SCHEME 8        1411
      COMMON /P/ IH0,IHE                                                SCHEME 9        1412
      COMMON /A/ MAXGRD                                                 SCHEME10        1413
      COMMON /C/ IWARN,LINE,KORIG,KNEW                                  SCHEME11        1414
      COMMON /BITS/ NBITIN,NBITEX,IPASS                                 SCHEME12        1415
      COMMON /TIME/ STIME,NCM                                           SCHEME13        1416
      COMMON /B/ IPARAM(20)                                             SCHEME14        1417
      COMMON /DOL/ ISTART(100),IGNORE(100),IFIRST(100)                  SCHEME15        1418
      COMMON /DOLL/ IDIM,ISTA,IIG,IFIR                                  SCHEME16        1419
      DIMENSION NODESL(100)                                             SCHEME17        1420
      EQUIVALENCE (IH,ATIME)                                            SCHEME18        1421
C     DETERMINE THE DEGREE OF EACH NODE.                                SCHEME19        1422
      CALL DEGREE(IG,II1,IDEG)                                          SCHEME20        1423
C DETERMINE MODD, THE MOST PREVALENT NODAL DEGREE.                      SCHEME21        1424
      MODD=MODE(IDEG,IPP)                                               SCHEME22        1425
C     DETERMINE THE NUMBER OF COMPONENTS, NCM.                          SCHEME23        1426
      NCM=COMPNT(IG,II1,IC,IDEG,IW,ICC)                                 SCHEME24        1427
C     DETERMINE THE MAXIMUM DEGREE OF ANY NODE.                         SCHEME25        1428
      MAXD=MAXDGR(0,IC,IDEG)                                            SCHEME26        1429
      MM=MAXD                                                           SCHEME27        1430
C     DETERMINE THE ORIGINAL BANDWIDTH,IS.                              SCHEME28        1431
      DO 30 I=1,NN                                                      SCHEME29        1432
      NEW(I)=I                                                          SCHEME30        1433
   30 ILD(I)=I                                                          SCHEME31        1434
      IS=MAXBND(0,IG,II1,IC,IDEG,NEW,ILD)                               SCHEME32        1435
      KORIG=IS                                                          SCHEME33        1436
      IH0=IH                                                            SCHEME34        1437
C     INITIALIZE  NEW  AND  ILD  ARRAYS.                                SCHEME35        1438
      DO 35 I=1,NN                                                      SCHEME36        1439
      NEW(I)=0                                                          SCHEME37        1440
   35 ILD(I)=0                                                          SCHEME38        1441
C     IF IP IS NOT EQUAL TO 0, THEN PRINT COMPONENT NUMBER,DEGREE,      SCHEME39        1442
C     AND CONNECTIONS FOR EACH NODE.                                    SCHEME40        1443
      IF(IP.EQ.0) GO TO 31                                              SCHEME41        1444
C PRINT INTERNAL NUMBER CONNECTION TABLE.                               SCHEME42        1445
      DO 60 I=1,NN                                                      SCHEME43        1446
      IF(MOD(I,LINE).EQ.1)WRITE(6,19)                                   SCHEME44        1447
   19 FORMAT(37H1LABEL  COMP MDIST  DEGR  CONNECTIONS ,10X,             SCHEME45        1448
     1   18H(INTERNAL NUMBERS) )                                        SCHEME46        1449
      MDIST=0                                                           SCHEME47        1450
      DO 65 J=1,MAXD                                                    SCHEME48        1451
      IS1=IUNPK(IG,MAXGRD*(J-1)+I,NBITIN)                               SCHEME49        1452
      IF(IS1.EQ.0)GO TO 65                                              SCHEME50        1453
      MDIST=MAXO(MDIST,IABS(I-IS1))                                     SCHEME51        1454
   65 CONTINUE                                                          SCHEME52        1455
      IPP(1)=IC(I)                                                      SCHEME53        1456
      IPP(2)=IDEG(I)                                                    SCHEME54        1457
      DO 610 IP1=1,MAXD                                                 SCHEME55        1458
  610 IPP(IP1+2)=IUNPK(IG,MAXGRD*(IP1-1)+I,NBITIN)                      SCHEME56        1459
      IS1=MAXD+2                                                        SCHEME57        1460
   60 WRITE(6,61)I,IPP(1),MDIST,(IPP(J),J=2,IS1)                        SCHEME58        1461
   61 FORMAT(5I6,20I5/ 25(25X,21I5/))                                   SCHEME59        1462
      WRITE(6,700)                                                      SCHEME60        1463
  700 FORMAT(1H1,//,32X,31HPROGRAMMER INFORMATION MESSAGES /)           SCHEME61        1464
      WRITE(6,29) IS,IH                                                 SCHEME62        1465
   29 FORMAT(19H ORIGINAL BANDWIDTH,I7,10H   PROFILE,I10)               SCHEME63        1466
      WRITE(6,27) MODD                                                  SCHEME64        1467
   27 FORMAT(30H MODE OF DEGREE DISTRIBUTION =,I5)                      SCHEME65        1468
      IF(ISTA.LE.0) GO TO 31                                            SCHEME66        1469
      WRITE(6,701)                                                      SCHEME67        1470
  701 FORMAT(34H STARTING NODES SUPPLIED BY USER -)                     SCHEME68        1471
      WRITE(6,100) (ISTART(I),I=1,ISTA)                                 SCHEME69        1472
C     TEST TIMER.                                                       SCHEME70        1473
   31 CALL SECOND(TBEG)                                                 SCHEME71        1474
      IF(IO.EQ.3) IS=IH                                                 SCHEME72        1475
C     GENERATE NUMBERING SCHEME FOR EACH COMPONENT, NC.                 SCHEME73        1476
      DO 500 NC=1,NCM                                                   SCHEME74        1477
C     DETERMINE THE RANGE OF DEGREES (MI  TO  MAD) OF NODES OF INTEREST.SCHEME75        1478
      MI=MINDEG(NC,IC,IDEG)                                             SCHEME76        1479
      MAD=MI                                                            SCHEME77        1480
      IF(NOM) 90,87,90                                                  SCHEME78        1481
   90 MA=MAXDGR(NC,IC,IDEG)                                             SCHEME79        1482
      MAD=MI+((MA-MI)*NUM)/NOM                                          SCHEME80        1483
C MAKE SURE THAT MAD IS LESS THAN MODD.                                 SCHEME81        1484
      MAD=MIN0(MAD,MODD-1)                                              SCHEME82        1485
      MAD=MAX0(MAD,MI)                                                  SCHEME83        1486
C     DETERMINE BANDWIDTH OR SUM CRITERION FOR EACH NODE MEETING SPECI- SCHEME84        1487
C              FIED CONDITION.                                          SCHEME85        1488
   87 IF(IP.EQ.0) GO TO 91                                              SCHEME86        1489
      WRITE(6,162) NC                                                   SCHEME87        1490
  162 FORMAT(22H *******      COMPONENT,I5,12H      *******)            SCHEME88        1491
      IF(IO.EQ.2) WRITE(6,169)                                          SCHEME89        1492
  169 FORMAT(43H OPTION 2 SELECTED  (CRITERION - BANDWIDTH ,            SCHEME90        1493
     +  57HMINIMIZATION;   CONDITION - MINMAX NUMBER OF NODES/LEVEL) )  SCHEME91        1494
      IF(IO.EQ.3) WRITE(6,179)                                          SCHEME92        1495
  179 FORMAT(52H OPTION 3 SELECTED (CRITERION - MINIMIZATION OF SUM;,    SCHEME93        1496
     +  44H   CONDITION - MINMAX NUMBER OF NODES/LEVEL) )               SCHEME94        1497
   91 CALL DIAM(NC,MAD,NL,NODESL,MAXLEV,IG,II1,IC,IDEG,IDIS,IW,ICC)     SCHEME95        1498
      IF(IP.EQ.0) GO TO 67                                              SCHEME96        1499
      WRITE(6,39) NC,MAD                                                SCHEME97        1500
```

```
      WRITE(6,59) MAXLEV                                                SCHEME98   1501
      WRITE(6,100) (NODESL(J),J=1,NL)                                   SCHEME99   1502
   67 CONTINUE                                                          SCHEM100   1503
      IF(ISTA.LE.0) GO TO 760                                           SCHEM101   1504
      M=0                                                               SCHEM102   1505
      DO 750 I=1,ISTA                                                   SCHEM103   1506
      J=ISTART(I)                                                       SCHEM104   1507
      IF(IC(J).NE.NC) GO TO 750                                         SCHEM105   1508
      M=M+1                                                             SCHEM106   1509
      DO 755 K=1,99                                                     SCHEM107   1510
      L=101-K                                                           SCHEM108   1511
  755 NODESL(L)=NODESL(L-1)                                             SCHEM109   1512
      NODESL(1)=J                                                       SCHEM110   1513
  750 CONTINUE                                                          SCHEM111   1514
      NL=MIN0(NL+M,100)                                                 SCHEM112   1515
      CALL FIXIT(NODESL,NL)                                             SCHEM113   1516
  760 CONTINUE                                                          SCHEM114   1517
      IF(IP.EQ.0) GO TO 63                                              SCHEM115   1518
      IF(ISTA.LE.0) GO TO 63                                            SCHEM116   1519
      WRITE(6,730)                                                      SCHEM117   1520
  730 FORMAT(48H MERGED LIST OF STARTING NODES SUPPLIED BY USER ,        SCHEM118   1521
     +  15HAND BY BANDIT -)                                            SCHEM119   1522
      WRITE(6,100) (NODESL(I),I=1,NL)                                   SCHEM120   1523
   39 FORMAT(10H COMPONENT,I5,19H    MAX DEGREE USED,I5)                SCHEM121   1524
   59 FORMAT(52H STARTING NODES FOR MINMAX NUMBER OF NODES PER LEVEL,I5)SCHEM122   1525
  100 FORMAT(4X,20I5)                                                   SCHEM123   1526
   63 CONTINUE                                                          SCHEM124   1527
      JMAX=MIN0(NT,NL)                                                  SCHEM125   1528
      IH=900000000                                                      SCHEM126   1529
      IHM=IH                                                            SCHEM127   1530
      DO 400 J=1,JMAX                                                   SCHEM128   1531
      CALL RELABL(1,NODESL(J ),IG,II1,IC,IDEG,IDIS,IW,NEW,ICC,ILD)      SCHEM129   1532
      IB=MAXBND(NC,IG,II1,IC,IDEG,NEW,ILD)                              SCHEM130   1533
      IF(IP.NE.0) WRITE(6,69) NODESL(J),IB,IH                           SCHEM131   1534
   69 FORMAT(14H STARTING NODE,I6,4X,9HBANDWIDTH,I6,3X,7HPROFILE,I8)    SCHEM132   1535
      IF(IO.EQ.3) IB=IH                                                 SCHEM133   1536
      IE=ICC(NC+1)-1                                                    SCHEM134   1537
      IF(IH-IB) 400,350,300                                             SCHEM135   1538
  300 IH=IB                                                             SCHEM136   1539
      IHM=IH                                                            SCHEM137   1540
      IJ=J                                                              SCHEM138   1541
      GO TO 400                                                         SCHEM139   1542
  350 IF(IHM.LE.IH) GO TO 400                                           SCHEM140   1543
      IHM=IH                                                            SCHEM141   1544
      IJ=J                                                              SCHEM142   1545
  400 CONTINUE                                                          SCHEM143   1546
      CALL RELABL(1,NODESL(IJ),IG,II1,IC,IDEG,IDIS,IW,NEW,ICC,ILD)      SCHEM144   1547
  500 CONTINUE                                                          SCHEM145   1548
      CALL STACK(IDEG,NEW,ILD,IW)                                       SCHEM146   1549
      IB=MAXBND(0,IG,II1,IC,IDEG,NEW,ILD)                               SCHEM147   1550
      IF(IP.EQ.0) GO TO 710                                             SCHEM148   1551
      WRITE(6,705)                                                      SCHEM149   1552
  705 FORMAT(21H0 ORIGINAL LABELING -)                                  SCHEM150   1553
      WRITE(6,708) KORIG,IH0                                            SCHEM151   1554
      WRITE(6,707)                                                      SCHEM152   1555
  707 FORMAT(21H  STD CM RELABELING -)                                  SCHEM153   1556
      WRITE(6,708) IB,IH                                                SCHEM154   1557
  708 FORMAT(1H+,26X,9HBANDWIDTH,I7,10X,7HPROFILE,I10)                  SCHEM155   1558
  709 FORMAT(21H  REV CM RELABELING -)                                  SCHEM156   1559
  710 IF(IO.EQ.3) IB=IH                                                 SCHEM157   1560
C PROFILE = SUM CRIT                                                    SCHEM158   1561
C IS=ORIGINAL BANDWIDTH (OR SUM CRIT IF IO.EQ.3)                        SCHEM159   1562
C IB=CURRENT BANDWIDTH (OR SUM CRIT IF IO.EQ.3)                         SCHEM160   1563
C IH=CURRENT PROFILE, IH0=ORIGINAL PROFILE                              SCHEM161   1564
      IF(IB-IS) 715,742,744                                             SCHEM162   1565
  742 IF(IH.LT.IH0) GO TO 715                                           SCHEM163   1566
  744 DO 712 I=1,NN                                                     SCHEM164   1567
      ILD(I)=I                                                          SCHEM165   1568
  712 NEW(I)=I                                                          SCHEM166   1569
      CALL STACK(IDEG,NEW,ILD,IW)                                       SCHEM167   1570
      IB=IS                                                             SCHEM168   1571
      IH=IH0                                                            SCHEM169   1572
      IF(IP.EQ.0) GO TO 715                                             SCHEM170   1573
      WRITE(6,713)                                                      SCHEM171   1574
  713 FORMAT(21H ORIG CM RELABELING -)                                  SCHEM172   1575
      WRITE(6,708) IB,IH                                                SCHEM173   1576
  715 IHE=IH                                                            SCHEM174   1577
      CALL REVERS(NEW,ILD)                                              SCHEM175   1578
      IB=MAXBND(0,IG,II1,IC,IDEG,NEW,ILD)                               SCHEM176   1579
      IF(IP.EQ.0) GO TO 717                                             SCHEM177   1580
      WRITE(6,709)                                                      SCHEM178   1581
      WRITE(6,708) IB,IH                                                SCHEM179   1582
  717 IF(IH.LT.IHE) GO TO 720                                           SCHEM180   1583
      CALL REVERS(NEW,ILD)                                              SCHEM181   1584
      IB=MAXBND(0,IG,II1,IC,IDEG,NEW,ILD)                               SCHEM182   1585
  720 IHE=IH                                                            SCHEM183   1586
      KNEW=IB                                                           SCHEM184   1587
      IF(IP.EQ.0) GO TO 508                                             SCHEM185   1588
      WRITE(6,722)                                                      SCHEM186   1589
  722 FORMAT(21H  ** FINAL LABELING -)                                  SCHEM187   1590
      WRITE(6,708) KNEW,IHE                                             SCHEM188   1591
  508 CALL SECOND(ATIME)                                                SCHEM189   1592
      ATIME=ATIME-TBEG                                                  SCHEM190   1593
      IF(IP.EQ.0) GO TO 600                                             SCHEM191   1594
      WRITE(6,89)ATIME                                                  SCHEM192   1595
   89 FORMAT(7H TIME =,F9.3,6H  SEC.)                                   SCHEM193   1596
  600 RETURN                                                            SCHEM194   1597
      END                                                               SCHEM195   1598
      SUBROUTINE STACK(IDEG,NEW,ILD,IW)                                 STACK  2   1599
C STACK POINTS OF ZERO DEGREE AT END OF THE NUMBERING.                  STACK  3   1600
```

```
      DIMENSION IDEG(1),NEW(1),ILD(1),IW(1)                          STACK  4      1601
C IW IS SCRATCH STORAGE.                                             STACK  5      1602
      COMMON /S/ NN                                                  STACK  6      1603
      COMMON /ZERO/ KT                                               STACK  7      1604
      KT=0                                                           STACK  8      1605
      NN1=NN-1                                                       STACK  9      1606
C LIST POINTS OF ZERO DEGREE AND INCREMENT COUNTER KT.               STACK 10      1607
      DO 10 I=1,NN                                                   STACK 11      1608
      IF(IDEG(I).GT.0) GO TO 10                                      STACK 12      1609
      KT=KT+1                                                        STACK 13      1610
      IW(KT)=ILD(I)                                                  STACK 14      1611
   10 CONTINUE                                                       STACK 15      1612
      IF(KT.LE.0) GO TO 70                                           STACK 16      1613
C SORT LIST OF RENUMBERED NUMBERS TO BE STACKED.                     STACK 17      1614
      CALL SORT(IW,KT)                                               STACK 18      1615
C STACK POINTS OF ZERO DEGREE AT END OF NEW.                         STACK 19      1616
      DO 40 L=1,KT                                                   STACK 20      1617
      I=IW(L)-L+1                                                    STACK 21      1618
      K=NEW(I)                                                       STACK 22      1619
      IF(I.GE.NN) GO TO 30                                           STACK 23      1620
      DO 20 J=I,NN1                                                  STACK 24      1621
   20 NEW(J)=NEW(J+1)                                                STACK 25      1622
   30 NEW(NN)=K                                                      STACK 26      1623
   40 CONTINUE                                                       STACK 27      1624
C CORRECT ILD, THE INVERSE OF NEW.                                   STACK 28      1625
   70 DO 80 I=1,NN                                                   STACK 29      1626
      K=NEW(I)                                                       STACK 30      1627
   80 ILD(K)=I                                                       STACK 31      1628
      RETURN                                                         STACK 32      1629
      END                                                            STACK 33      1630
      SUBROUTINE REVERS(NEW,ILD)                                     REVERS 2      1631
C REVERSE THE NUMBERING OF THE FIRST NN-KT GRID POINTS.              REVERS 3      1632
C NN=NUMBER OF GRID POINTS.                                          REVERS 4      1633
C KT=THE NUMBER OF POINTS OF ZERO DEGREE (STACKED AT END OF NEW      REVERS 5      1634
C       BY STACK)                                                    REVERS 6      1635
      DIMENSION NEW(1),ILD(1)                                        REVERS 7      1636
      COMMON /S/ NN                                                  REVERS 8      1637
      COMMON /ZERO/ KT                                               REVERS 9      1638
C REVERSE NEW ARRAY.                                                 REVERS10      1639
      J=(NN-KT)/2                                                    REVERS11      1640
      LL=NN-KT+1                                                     REVERS12      1641
      DO 10 I=1,J                                                    REVERS13      1642
      L=LL-I                                                         REVERS14      1643
      K=NEW(L)                                                       REVERS15      1644
      NEW(L)=NEW(I)                                                  REVERS16      1645
   10 NEW(I)=K                                                       REVERS17      1646
C CORRECT ILD, THE INVERSE OF NEW.                                   REVERS18      1647
      DO 20 I=1,NN                                                   REVERS19      1648
      K=NEW(I)                                                       REVERS20      1649
   20 ILD(K)=I                                                       REVERS21      1650
      RETURN                                                         REVERS22      1651
      END                                                            REVERS23      1652
      SUBROUTINE DEGREE(IG,II1,IDEG)                                 DEGREE 2      1653
C SET UP THE IDEG ARRAY CONTAINING THE DEGREE OF EACH NODE STORED    DEGREE 3      1654
C     IN THE IG ARRAY.                                               DEGREE 4      1655
C   IDEG(I)=DEGREE OF NODE I                                         DEGREE 5      1656
      DIMENSION IG(II1,1),IDEG(1)                                    DEGREE 6      1657
      COMMON /S/ NN,MM,IH,IB                                         DEGREE 7      1658
      COMMON /A/ MAXGRD                                              DEGREE 8      1659
      COMMON /BITS/ NBITIN,NBITEX,IPASS                              DEGREE 9      1660
      DO 100 I=1,NN                                                  DEGREE10      1661
      IDEG(I)=0                                                      DEGREE11      1662
      DO 80 J=1,MM                                                   DEGREE12      1663
      IF(IUNPK(IG,MAXGRD*(J-1)+I,NBITIN)) 100,100,50                 DEGREE13      1664
   50 IDEG(I)=IDEG(I)+1                                              DEGREE14      1665
   80 CONTINUE                                                       DEGREE15      1666
  100 CONTINUE                                                       DEGREE16      1667
      RETURN                                                         DEGREE17      1668
      END                                                            DEGREE18      1669
      FUNCTION MODE(IDEG,MODD)                                       MODE   2      1670
C COMPUTE MODE, THE MOST PREVALENT NODAL DEGREE.    IF SEVERAL DEGREES MODE   3      1671
C     ARE EQUALLY PREVALENT, THE LOWEST IS CHOSEN.                   MODE   4      1672
      COMMON /S/ NN,MM                                               MODE   5      1673
      DIMENSION IDEG(1),MODD(1)                                      MODE   6      1674
C IDEG(I)=DEGREE OF NODE I                                           MODE   7      1675
C  MODD(I)=NUMBER OF NODES OF DEGREE I                               MODE   8      1676
      DO 10 I=1,MM                                                   MODE   9      1677
   10 MODD(I)=0                                                      MODE  10      1678
      DO 20 I=1,NN                                                   MODE  11      1679
      K=IDEG(I)                                                      MODE  12      1680
   20 MODD(K)=MODD(K)+1                                              MODE  13      1681
      MODE=0                                                         MODE  14      1682
      MAX=0                                                          MODE  15      1683
      DO 30 I=1,MM                                                   MODE  16      1684
      K=MODD(I)                                                      MODE  17      1685
      IF(K.LE.MAX) GO TO 30                                          MODE  18      1686
      MAX=K                                                          MODE  19      1687
      MODE=I                                                         MODE  20      1688
   30 CONTINUE                                                       MODE  21      1689
      RETURN                                                         MODE  22      1690
      END                                                            MODE  23      1691
      FUNCTION COMPNT(IG,II1,IC,IDEG,IW,ICC)                         COMPNT 2      1692
C THIS FUNCTION HAS AS ITS VALUE THE NUMBER OF COMPONENTS STORED     COMPNT 3      1693
C     IN THE CONNECTION ARRAY IG.                                    COMPNT 4      1694
C ALSO, IC AND ICC ARE SET UP.                                       COMPNT 5      1695
C   IC(I)=COMPONENT INDEX FOR NODE I                                 COMPNT 6      1696
C   ICC(I)=THE STARTING POSITION TO BE USED FOR LABELS IN COMPONENT I COMPNT 7      1697
C THUS, ICC(I+1)-ICC(I)= THE NUMBER OF NODES IN COMPONENT I          COMPNT 8      1698
      DIMENSION IG(II1,1),IC(1),IDEG(1),IW(1),ICC(1)                 COMPNT 9      1699
      COMMON /S/ NN,MM,IH,IB                                         COMPNT10      1700
```

```
      COMMON /A/ MAXGRD                                                 COMPNT11  1701
      COMMON /BITS/ NBITIN,NBITEX,IPASS                                 COMPNT12  1702
C INITIALIZE ARRAYS.                                                    COMPNT13  1703
      DO 100 I=1,NN                                                     COMPNT14  1704
      ICC(I)=0                                                          COMPNT15  1705
      IC(I)=0                                                           COMPNT16  1706
  100 CONTINUE                                                          COMPNT17  1707
      NC=0                                                              COMPNT18  1708
      ICC(1)=1                                                          COMPNT19  1709
C CHECK IF IC IS COMPLETE.                                              COMPNT20  1710
  105 DO 110 I=1,NN                                                     COMPNT21  1711
      IF(IC(I)) 110,120,110                                             COMPNT22  1712
  110 COMPNT=NC                                                         COMPNT23  1713
      RETURN                                                            COMPNT24  1714
  120 NC=NC+1                                                           COMPNT25  1715
      KI=0                                                              COMPNT26  1716
      KO=1                                                              COMPNT27  1717
      IW(1)=I                                                           COMPNT28  1718
      IC(I)=NC                                                          COMPNT29  1719
      IF(NC-1)130,125,125                                               COMPNT30  1720
125   IS=ICC(NC)+1                                                      COMPNT31  1721
      ICC(NC+1)=IS                                                      COMPNT32  1722
  130 KI=KI+1                                                           COMPNT33  1723
      II=IW(KI)                                                         COMPNT34  1724
      N=IDEG(II)                                                        COMPNT35  1725
      IF(N)140,105,140                                                  COMPNT36  1726
  140 DO 200 I=1,N                                                      COMPNT37  1727
      IA=IUNPK(IG,MAXGRD*(I-1)+II,NBITIN)                               COMPNT38  1728
      IF(IC(IA)) 200,150,200                                            COMPNT39  1729
150   IC(IA)=NC                                                         COMPNT40  1730
      KO=KO+1                                                           COMPNT41  1731
      IW(KO)=IA                                                         COMPNT42  1732
      IS=ICC(NC+1)+1                                                    COMPNT43  1733
      ICC(NC+1)=IS                                                      COMPNT44  1734
  200 CONTINUE                                                          COMPNT45  1735
      IF(KO-KI)105,105,130                                              COMPNT46  1736
      END                                                               COMPNT47  1737
      FUNCTION MAXDGR(NC,IC,IDEG)                                       MAXDGR 2  1738
C THIS FUNCTION HAS AS ITS VALUE THE MAXIMUM DEGREE OF ANY NODE OF      MAXDGR 3  1739
C       COMPONENT NC IF NC.GT.0                                         MAXDGR 4  1740
C IF NC.LE.0, ALL COMPONENTS ARE CONSIDERED.                            MAXDGR 5  1741
      DIMENSION IC(1),IDEG(1)                                           MAXDGR 6  1742
      COMMON /S/ NN,MM,IH,IB                                            MAXDGR 7  1743
      M=0                                                               MAXDGR 8  1744
      DO 100 I=1,NN                                                     MAXDGR 9  1745
      IF(NC)40,50,40                                                    MAXDGR10  1746
40    IF(IC(I)-NC) 100,50,100                                           MAXDGR11  1747
50    IF(IDEG(I)-M) 100,100,60                                          MAXDGR12  1748
60    M=IDEG(I)                                                         MAXDGR13  1749
  100 CONTINUE                                                          MAXDGR14  1750
      MAXDGR=M                                                          MAXDGR15  1751
      RETURN                                                            MAXDGR16  1752
      END                                                               MAXDGR17  1753
      FUNCTION MAXBND(NC,IG,II1,IC,IDEG,NEW,ILD)                        MAXBND 2  1754
C THIS FUNCTION HAS AS ITS VALUE THE MAXIMUM DIFFERENCE BETWEEN NODE    MAXBND 3  1755
C     LABELS OF CONNECTED NODES FOR NODES OF COMPONENT NC.GT.0          MAXBND 4  1756
C IF NC.LE.0, ALL COMPONENTS ARE CONSIDERED.                            MAXBND 5  1757
C THE NODAL RENUMBERING DEFINED BY ILD AND NEW MUST BE SET UP PRIOR     MAXBND 6  1758
C       TO THE FUNCTION CALL.                                           MAXBND 7  1759
C COMPUTE IH, THE SUM CRIT (PROFILE).                                   MAXBND 8  1760
      DIMENSION IG(II1,1),IC(1),IDEG(1),NEW(1),ILD(1)                   MAXBND 9  1761
      COMMON /S/ NN,MM,IH,IB                                            MAXBND10  1762
      COMMON /A/ MAXGRD                                                 MAXBND11  1763
      COMMON /BITS/ NBITIN,NBITEX,IPASS                                 MAXBND12  1764
      IH=0                                                              MAXBND13  1765
      M=0                                                               MAXBND14  1766
      DO 100 I=1,NN                                                     MAXBND15  1767
      MX=0                                                              MAXBND16  1768
      IA=NEW(I)                                                         MAXBND17  1769
      IF(NC)40,50,40                                                    MAXBND18  1770
40    IF(IA.EQ.0)GO TO 100                                              MAXBND19  1771
      IF(NC-IC(IA)) 100,50,100                                          MAXBND20  1772
50    N=IDEG(IA)                                                        MAXBND21  1773
      IF(N)100,100,150                                                  MAXBND22  1774
150   DO 90 J=1,N                                                       MAXBND23  1775
      II=IUNPK(IG,MAXGRD*(J-1)+IA,NBITIN)                               MAXBND24  1776
      IB=MAX0(0,I-ILD(II))                                              MAXBND25  1777
      IF(IB.GT.MX) MX=IB                                                MAXBND26  1778
90    CONTINUE                                                          MAXBND27  1779
      IF(MX.GT.M) M=MX                                                  MAXBND28  1780
      IH=IH+MX                                                          MAXBND29  1781
100   CONTINUE                                                          MAXBND30  1782
      MAXBND=M                                                          MAXBND31  1783
      RETURN                                                            MAXBND32  1784
      END                                                               MAXBND33  1785
      FUNCTION MINDEG(NC,IC,IDEG)                                       MINDEG 2  1786
C THIS FUNCTION HAS AS ITS VALUE THE MINIMUM DEGREE OF ANY NODE OF      MINDEG 3  1787
C       COMPONENT NC IF NC.GT.0                                         MINDEG 4  1788
C IF NC.LE.0, ALL COMPONENTS ARE CONSIDERED.                            MINDEG 5  1789
      DIMENSION IC(1),IDEG(1)                                           MINDEG 6  1790
      COMMON /S/ NN,MM,IH,IB                                            MINDEG 7  1791
      M=10000                                                           MINDEG 8  1792
      DO 100 I=1,NN                                                     MINDEG 9  1793
      IF(NC)40,50,40                                                    MINDEG10  1794
40    IF(IC(I)-NC) 100,50,100                                           MINDEG11  1795
50    IF(M-IDEG(I)) 100,100,60                                          MINDEG12  1796
60    M=IDEG(I)                                                         MINDEG13  1797
  100 CONTINUE                                                          MINDEG14  1798
      MINDEG=M                                                          MINDEG15  1799
      RETURN                                                            MINDEG16  1800
```

```
      END                                                               MINDEG17        1801
      SUBROUTINE DIAM(NC,MAXDEG,NL,NODESL,MAXLEV,                       DIAM   2        1802
     + IG,II1,IC,IDEG,IDIS,IW,ICC)                                      DIAM   3        1803
C DETERMINE NL STARTING POINTS AND STORE IN NODESL.                     DIAM   4        1804
      DIMENSION IG(II1,1),IDIS(1),IW(1),ICC(1),IC(1),IDEG(1)            DIAM   5        1805
      COMMON /S/ NN,MM,IH,I8                                            DIAM   6        1806
      COMMON /A/ MAXGRD                                                 DIAM   7        1807
      COMMON /BITS/ NBITIN,NBITEX,IPASS                                 DIAM   8        1808
      DIMENSION NODESL(1)                                               DIAM   9        1809
      NL=0                                                              DIAM  10        1810
      MAXLEV=10000                                                      DIAM  11        1811
      DO 100 I=1,NN                                                     DIAM  12        1812
      IF(NC-IC(I)) 100,40,100                                           DIAM  13        1813
40    IF(MAXDEG-IDEG(I)) 100,105,105                                    DIAM  14        1814
  105 MD=IDIST(I,ML,MAXLEV,IG,II1,IC,IDEG,IDIS,IW,ICC)                  DIAM  15        1815
      IF(MD) 115,115,56                                                 DIAM  16        1816
   56 IF(ML-MAXLEV)58,64,100                                            DIAM  17        1817
   58 MAXLEV=ML                                                         DIAM  18        1818
      NL=1                                                              DIAM  19        1819
      NODESL(1)=I                                                       DIAM  20        1820
      GO TO 100                                                         DIAM  21        1821
   64 IF(NL.GE.100) GO TO 100                                           DIAM  22        1822
      NL=NL+1                                                           DIAM  23        1823
      NODESL(NL)=I                                                      DIAM  24        1824
  100 CONTINUE                                                          DIAM  25        1825
  110 RETURN                                                            DIAM  26        1826
  115 ML=1                                                              DIAM  27        1827
      NODESL(1)=I                                                       DIAM  28        1828
      MAXLEV=0                                                          DIAM  29        1829
      RETURN                                                            DIAM  30        1830
      END                                                               DIAM  31        1831
      SUBROUTINE RELABL(NS,NODES,IG,II1,IC,IDEG,IDIS,IW,NEW,ICC,ILD)    RELABL 2        1832
C GENERATE A RELABELING SCHEME STARTING WITH NS NODES FOR WHICH         RELABL 3        1833
C       LABELS HAVE BEEN STORED IN ARRAY NODES.                         RELABL 4        1834
C SET UP ILD AND NEW.                                                   RELABL 5        1835
C      ILD(OLD)=NEW                                                     RELABL 6        1836
C       NEW(NEW)=OLD,   THE INVERSE OF ILD                              RELABL 7        1837
      DIMENSION IG(II1,1),IC(1),IDEG(1),IDIS(1),IW(1),NEW(1),ICC(1)     RELABL 8        1838
      DIMENSION ILD(1)                                                  RELABL 9        1839
      COMMON /S/ NN,MM,IH,I8                                            RELABL10        1840
      INTEGER X                                                         RELABL11        1841
      COMMON /A/ MAXGRD                                                 RELABL12        1842
      COMMON /BITS/ NBITIN,NBITEX,IPASS                                 RELABL13        1843
      DIMENSION NODES( 1),IAJ(50)                                       RELABL14        1844
      I=NODES(1)                                                        RELABL15        1845
      ICN=IC(I)                                                         RELABL16        1846
      NT=ICC(ICN)-1                                                     RELABL17        1847
      DO 50 I=1,NN                                                      RELABL18        1848
      IF(IC(I)-ICN) 50,40,50                                            RELABL19        1849
40    IDIS(I)=0                                                         RELABL20        1850
50    CONTINUE                                                          RELABL21        1851
      DO 100 J=1,NS                                                     RELABL22        1852
      JJ=NODES(J)                                                       RELABL23        1853
      IDIS(JJ)=-1                                                       RELABL24        1854
      JT=J+NT                                                           RELABL25        1855
      NEW(JT)=JJ                                                        RELABL26        1856
100   ILD(JJ)=JT                                                       RELABL27        1857
      KI=NT                                                             RELABL28        1858
      KO=NS+NT                                                          RELABL29        1859
      LL=KO                                                             RELABL30        1860
      L=1                                                               RELABL31        1861
      J=KO                                                              RELABL32        1862
      NNC=ICC(ICN+1)-1                                                  RELABL33        1863
130   KI=KI+1                                                           RELABL34        1864
      IF(KI-LL)135,132,135                                              RELABL35        1865
132   L=L+1                                                             RELABL36        1866
      LL=KO+1                                                           RELABL37        1867
135   II=NEW(KI)                                                        RELABL38        1868
      N=IDEG(II)                                                        RELABL39        1869
      IF(N)140,255,140                                                  RELABL40        1870
140   IJ=0                                                              RELABL41        1871
      DO 200 I=1,N                                                      RELABL42        1872
      IA=IUNPK(IG,MAXGRD*(I-1)+II,NBITIN)                               RELABL43        1873
      IF(IDIS(IA)) 200,150,200                                          RELABL44        1874
150   IJ=IJ+1                                                           RELABL45        1875
      IDIS(IA)=L                                                        RELABL46        1876
      KO=KO+1                                                           RELABL47        1877
      IAJ(IJ)=IA                                                        RELABL48        1878
      IW(IJ)=IDEG(IA)                                                   RELABL49        1879
200   CONTINUE                                                          RELABL50        1880
      IF(IJ-1)250,210,220                                               RELABL51        1881
210   J=KO                                                              RELABL52        1882
      IZ=IAJ(1)                                                         RELABL53        1883
      NEW(KO)=IZ                                                        RELABL54        1884
      ILD(IZ)=KO                                                        RELABL55        1885
      GO TO 250                                                         RELABL56        1886
220   X=0                                                               RELABL57        1887
221   DO 230 I=2,IJ                                                     RELABL58        1888
      IF(IW(I)-IW(I-1))224,230,230                                      RELABL59        1889
224   CONTINUE                                                          RELABL60        1890
      X=IW(I)                                                           RELABL61        1891
      IW(I)=IW(I-1)                                                     RELABL62        1892
      IW(I-1)=X                                                         RELABL63        1893
225   X=IAJ(I)                                                          RELABL64        1894
      IAJ(I)=IAJ(I-1)                                                   RELABL65        1895
      IAJ(I-1)=X                                                        RELABL66        1896
230   CONTINUE                                                          RELABL67        1897
      IF(X)235,235,220                                                  RELABL68        1898
235   DO 240 I=1,IJ                                                     RELABL69        1899
      J=J+1                                                             RELABL70        1900
```

```
      IZ=IAJ(I)                                                          RELABL71   1901
      NEW(J)=IZ                                                          RELABL72   1902
      ILD(IZ)=J                                                          RELABL73   1903
240   CONTINUE                                                           RELABL74   1904
250   IF(KO-NNC)130,255,255                                              RELABL75   1905
255   CONTINUE                                                           RELABL76   1906
      RETURN                                                             RELABL77   1907
      END                                                                RELABL78   1908
      FUNCTION IDIST(NS,ML,MAXLEV,IG,III,IC,IDEG,IDIS,IW,ICC)            IDIST   2  1909
C THIS FUNCTION HAS AS ITS VALUE THE MAXIMUM DISTANCE OF ANY NODE        IDIST   3  1910
C      IN COMPONENT IC(NS) FROM THE NODE NS.                             IDIST   4  1911
C THE DISTANCE OF EACH NODE IN THIS COMPONENT IS STORED IN THE ARRAY      IDIST   5  1912
C          IDIS.                                                         IDIST   6  1913
C THE MAXIMUM NUMBER OF NODES AT THE SAME DISTANCE FROM NS IS            IDIST   7  1914
C       STORED IN ML.                                                    IDIST   8  1915
      DIMENSION IG(III,1),IC(1),IDEG(1),IDIS(1),IW(1),ICC(1)             IDIST   9  1916
      COMMON /S/ NN,MM,IH,IB                                             IDIST  10  1917
      COMMON /A/ MAXGRD                                                  IDIST  11  1918
      COMMON /BITS/ NBITIN,NBITEX,IPASS                                  IDIST  12  1919
      ICN=IC(NS)                                                         IDIST  13  1920
      NNC=ICC(ICN+1)-ICC(ICN)                                            IDIST  14  1921
      DO 50 I=1,NN                                                       IDIST  15  1922
      IF(IC(I)-IC(NS)) 50,40,50                                          IDIST  16  1923
40    IDIS(I)=0                                                          IDIST  17  1924
   50 CONTINUE                                                           IDIST  18  1925
      LL=1                                                               IDIST  19  1926
      L=0                                                                IDIST  20  1927
      KI=0                                                               IDIST  21  1928
      KO=1                                                               IDIST  22  1929
      ML=0                                                               IDIST  23  1930
      IW(1)=NS                                                           IDIST  24  1931
      IDIS(NS)=-1                                                        IDIST  25  1932
  130 KI=KI+1                                                            IDIST  26  1933
      IF(KI-LL)135,132,135                                               IDIST  27  1934
  132 L=L+1                                                              IDIST  28  1935
      LL=KO+1                                                            IDIST  29  1936
      K=KO-KI+1                                                          IDIST  30  1937
      IF(K-ML) 135,135,133                                               IDIST  31  1938
  133 ML=K                                                               IDIST  32  1939
      IF(ML-MAXLEV) 135,135,220                                          IDIST  33  1940
135    II=IW(KI)                                                         IDIST  34  1941
      N=IDEG(II)                                                         IDIST  35  1942
      IF(N)140,215,140                                                   IDIST  36  1943
  140 DO 200 I=1,N                                                       IDIST  37  1944
      IA=IUNPK(IG,MAXGRD*(I-1)+II,NBITIN)                                IDIST  38  1945
      IF(IDIS(IA))200,150,200                                            IDIST  39  1946
150   IDIS(IA)=L                                                         IDIST  40  1947
      KO=KO+1                                                            IDIST  41  1948
      IW(KO)=IA                                                          IDIST  42  1949
  200 CONTINUE                                                           IDIST  43  1950
      IF(KO-NNC)130,205,205                                              IDIST  44  1951
  205 IDIST=L                                                            IDIST  45  1952
      IDIS(NS)=0                                                         IDIST  46  1953
      K=KO-KI                                                            IDIST  47  1954
      IF(K-ML) 206,206,207                                               IDIST  48  1955
  207 ML=K                                                               IDIST  49  1956
  206 CONTINUE                                                           IDIST  50  1957
      RETURN                                                             IDIST  51  1958
  215 L=0                                                                IDIST  52  1959
      GO TO 205                                                          IDIST  53  1960
  220 IDIST=1                                                            IDIST  54  1961
      RETURN                                                             IDIST  55  1962
      END                                                                IDIST  56  1963
          IDENT  PCUP                                                    PCUP    2  1964
          LIST   D,M,A                                                   PCUP    3  1965
          ENTRY  PACK,IPACK,UNPK,IUNPK,ABT                               PCUP    4  1966
          EXT    CPC                                                     PCUP    5  1967
          USE    /BITS/                                                  PCUP    6  1968
         BSS     2                                                       PCUP    7  1969
IPASS     BSS    1                                                       PCUP    8  1970
          USE    0                                                       PCUP    9  1971
SIXTY     DATA   60.                                                     PCUP   10  1972
 PACK     BSSZ   1                                                       PCUP   11  1973
 IPACK    EQU    PACK                                                    PCUP   12  1974
*  SET FLAG TO INDICATE A PACK INSTRUCTION REQUIRED                      PCUP   13  1975
          SB1    1                                                       PCUP   14  1976
*  LOAD A0 WITH THE ADDRESS OF THE ARGUMENT LIST                         PCUP   15  1977
 INIT     SB7    A0                                                      PCUP   16  1978
          SA0    A1                                                      PCUP   17  1979
*  INCREASE ITERATION INDEX BY ONE                                       PCUP   18  1980
          SA5    IPASS                                                   PCUP   19  1981
          SX6    1                                                       PCUP   20  1982
          IX6    X5+X6                                                   PCUP   21  1983
          SA6    A5+B0                                                   PCUP   22  1984
*  LOAD X6  WITH THE ADDRESS OF ARGUMENT LIST FOR LATER LOADING OF       PCUP   23  1985
*  PACKED WORD                                                           PCUP   24  1986
          BX6    X1                                                      PCUP   25  1987
*  LOAD X1 WITH THE SUBSCRIPT OF THE ARRAY                               PCUP   26  1988
          SA1    A0+1                                                    PCUP   27  1989
          SA1    X1                                                      PCUP   28  1990
*  CONVERT SUBSCRIPT FROM INTEGER TO REAL                                PCUP   29  1991
          PX1    B0,X1                                                   PCUP   30  1992
          NX1    X1                                                      PCUP   31  1993
*  LOAD X2 WITH THE NUMBER OF BITS PER WORD                              PCUP   32  1994
          SA2    SIXTY                                                   PCUP   33  1995
*  LOAD X3 WITH THE NUMBER OF BITS TO BE DEVOTED TO EACH PACKED WORD     PCUP   34  1996
          SA3    A0+2                                                    PCUP   35  1997
          SA3    X3                                                      PCUP   36  1998
*  LOAD B6 WITH THE NUMBER OF BITS IN WORD NOT DEVOTED TO THE PACKED     PCUP   37  1999
*  WORD                                                                  PCUP   38  2000
```

```
          SB6     X3-60                                          PCUP  39        2001
          S96     -B6                                            PCUP  40        2002
*  CONVERT THE NUMBER OF BITS PER PACKED WORD FROM INTEGER TO REAL PCUP  41        2003
          PX3     B0,X3                                          PCUP  42        2004
          NX3     X3                                             PCUP  43        2005
*  LOAD X2 WITH THE NUMBER OF PACKED WORDS THAT CAN EXIST PER 60 BIT PCUP  44        2006
*  WORD                                                           PCUP  45        2007
          FX2     X2/X3                                          PCUP  46        2008
*  TRUNCATE X2 TO LOSE ANY FRACTIONAL PART                        PCUP  47        2009
          UX2     B2,X2                                          PCUP  48        2010
          LX2     B2,X2                                          PCUP  49        2011
          PX2     B0,X2                                          PCUP  50        2012
          NX2     X2                                             PCUP  51        2013
*  LOAD X5 WITH THE NUMBER OF 60 BIT WORDS NECESSARY TO LOCATE THE PCUP  52        2014
*  POSITION IN ABSOLUTE CORE THAT THE VARIABLE ADDRESSED          PCUP  53        2015
          FX5     X1/X2                                          PCUP  54        2016
*  CONVERT X5 TO INTEGER BRIEFLY FOR AN INTEGER ADD OPERATION     PCUP  55        2017
          UX5     B2,X5                                          PCUP  56        2018
          LX5     B2,X5                                          PCUP  57        2019
*  X6 NOW POINTS TO THE ABSOLUTE LOCATION IN CORE CONTAINING THE  PCUP  58        2020
*  VARIABLE DESIRED                                               PCUP  59        2021
          IX6     X5+X6                                          PCUP  60        2022
          PX5     B0,X5                                          PCUP  61        2023
          NX5     X5                                             PCUP  62        2024
*  LOAD X4 WITH THE NUMBER OF WHOLE 60 BIT WORDS TO FIND THE LOCATION PCUP  63        2025
*  DESIRED                                                        PCUP  64        2026
          FX4     X2*X5                                          PCUP  65        2027
*  X4 NOW CONTAINS OFFSET WITHIN THE LOCATION FOR PACKED VALUE    PCUP  66        2028
          FX4     X1-X4                                          PCUP  67        2029
          NX4     X4                                             PCUP  68        2030
*  CONVERT X4 TO INTEGER BRIEFLY FOR COMPARISON PURPOSES          PCUP  69        2031
          UX4     B2,X4                                          PCUP  70        2032
          LX4     B2,X4                                          PCUP  71        2033
*  SET X6 TO ITSELF - 1 FOR A ZERO X4 VALUE                       PCUP  72        2034
*  X4 = 0 INDICATES THAT THE PACKED VARIABLE ENDS ON A WORD BOUNDARY PCUP  73        2035
          NZ      X4,BR                                          PCUP  74        2036
          SX6     X6-1                                           PCUP  75        2037
*  LOAD X1 WITH THE NUMBER OF BITS TO BE SHIFTED                  PCUP  76        2038
 BR       PX4     B0,X4                                          PCUP  77        2039
          NX4     X4                                             PCUP  78        2040
          FX1     X3*X4                                          PCUP  79        2041
*  CONVERT X1 TO INTEGER AND SAVE IN B2 FOR LATER USE             PCUP  80        2042
          UX1     B2,X1                                          PCUP  81        2043
          LX1     B2,X1                                          PCUP  82        2044
          SB2     X1                                             PCUP  83        2045
*  LOAD THE VARIABLE OF CORE CONCERNED                            PCUP  84        2046
          SA2     X6                                             PCUP  85        2047
*  SHIFT THE WORD TO ALIGN PROPERLY FOR MASK                      PCUP  86        2048
          LX2     B2,X2                                          PCUP  87        2049
*  COMPLEMENT B2                                                  PCUP  88        2050
          SB2     -B2                                            PCUP  89        2051
*  FORM A 59 BIT MASK IN THE LOWER 59 BITS OF X4                  PCUP  90        2052
          MX4     59                                             PCUP  91        2053
          LX4     59                                             PCUP  92        2054
*  B2 NOW HAS THE NUMBER OF BITS PER PACKED WORD                  PCUP  93        2055
          SB2     B2+60                                          PCUP  94        2056
*  LOAD X4 WITH A MASK OF 1'S WHOSE LENGTH EQUALS THE THIRD ARGUMENT - 1PCUP  95        2057
          AX4     B6,X4                                          PCUP  96        2058
          SX5     1                                              PCUP  97        2059
*  X6 HAS A MASK FOR SIGN AND VALUE                               PCUP  98        2060
          IX5     X4+X5                                          PCUP  99        2061
          IX6     X4+X5                                          PCUP 100        2062
*  +0 TO UPK IF AN UNPACK IS DESIRED                              PCUP 101        2063
          EQ      B1,UPK                                         PCUP 102        2064
*  LOAD X1 WITH THE VALUE TO BE LOADED INTO CORE                  PCUP 103        2065
          SA1     A0+3                                           PCUP 104        2066
          SA1     X1                                             PCUP 105        2067
*  INSURE THAT VALUE IS NOT TOO LARGE FOR PROPER PACKING          PCUP 106        2068
          BX0     X1                                             PCUP 107        2069
          PL      X0,CHK                                         PCUP 108        2070
          BX0     -X0                                            PCUP 109        2071
CHK       IX0     X0-X4                                          PCUP 110        2072
          PL      X0,OMP                                         PCUP 111        2073
*  IF VALUE IS NEGATIVE, SET SIGN MASK TO ZERO                    PCUP 112        2074
          NG      X1,SCH                                         PCUP 113        2075
          SX5     B0                                             PCUP 114        2076
*  AND OUT THE VALUE TO BE PACKED INTO X1                         PCUP 115        2077
 SCH      BX1     X1*X4                                          PCUP 116        2078
*  OR THE SIGS INTO X1                                            PCUP 117        2079
          BX1     X1+X5                                          PCUP 118        2080
*  COMPLEMENT X6                                                  PCUP 119        2081
          BX6     -X6                                            PCUP 120        2082
*  ZERO OUT THE BITS INTO WHICH THE PACKED VALUE IS TO BE INSERTED PCUP 121        2083
          BX2     X2*X6                                          PCUP 122        2084
*  OR THE PACKED VALUE INTO A 60 BIT WORD AT THE PROPER LOCATION  PCUP 123        2085
          BX2     X1+X2                                          PCUP 124        2086
*  SHIFT THE 60 BIT WORD TO REALIGN WITH CORE                     PCUP 125        2087
          LX6     B2,X2                                          PCUP 126        2088
*  STORE THE 60 BIT PACKED WORD IN CORE                           PCUP 127        2089
*  THE FUNCTION RETURNS A 60 BIT PACKED VALUE                     PCUP 128        2090
          SA6     A2                                             PCUP 129        2091
*  RESTORE A0                                                     PCUP 130        2092
          SA0     B7                                             PCUP 131        2093
*  BRANCH OUT OF THE ROUTINE                                      PCUP 132        2094
          ZR      PACK                                           PCUP 133        2095
 UNPK     BSSZ    1                                              PCUP 134        2096
 IUNPK    EQU     UNPK                                           PCUP 135        2097
*  SET FLAG TO INDICATE AN  UNPACK INSTRUCTION IS DESIRED         PCUP 136        2098
          SB1     B0                                             PCUP 137        2099
          ZR      INIT                                           PCUP 138        2100
```

```
*  OR THE 60 BIT WORD OF CORE TO REMOVE 1 PACKED VALUE INTO X6
 UPK      BX6    X2*X4
*  OR A 60 BIT WORD TO REMOVE SIGN OF PACKED WORD INTO X5
          BX5    X5*X2
*  COMPLEMENT VALUE IF SIGN MASK INDICATES A NEGATIVE VALUE
          ZR     X5,SC
          BX4    -X4
          BX6    X4+X6
*  RESTORE A0
 SC       SA0    B7
*  BRANCH OUT OF THE ROUTINE
          ZR     UNPK
*  GENERATE A DAYFILE MESSAGE AND ABORT JOB IF AN INTEGER TO BE
*  PACKED IS TOO LARGE FOR THE SPECIFIED BIT PATTERN
PKMSS     DATA   C*VALUE TOO LARGE TO PACK*
DMP       MESSAGE PKMSS,,1
AB        ABORT
*  ENTRY POINT TO ABORT JOB AND EXECUTE EXIT CONTROL CARDS.
ABT       DATA   0
          EQ     AB
          END
          IDENT  CORS
          ENTRY  CORSIZ
          VFD    36/6HCORSIZ,24/0
          USE    //
A         BSS    1
          USE    /K/
          BSS     7
KORE      BSS    1
CM        BSS    1
          USE    0
CORSIZ    DATA   0
          MEMORY CM,STATUS,1
          SX0    1B
          SA3    STATUS
          IX6    X3-X0
          LX6    30
          SA6    CM
          SX5    A
          IX6    X6-X5
          SA6    KORE
          EQ     CORSIZ
STATUS    DATA   0
          END
```

```
PCUP 139    2101
PCUP 140    2102
PCUP 141    2103
PCUP 142    2104
PCUP 143    2105
PCUP 144    2106
PCUP 145    2107
PCUP 146    2108
PCUP 147    2109
PCUP 148    2110
PCUP 149    2111
PCUP 150    2112
PCUP 151    2113
PCUP 152    2114
PCUP 153    2115
PCUP 154    2116
PCUP 155    2117
PCUP 156    2118
PCUP 157    2119
PCUP 158    2120
PCUP 159    2121
CORS   2    2122
CORS   3    2123
CORS   4    2124
CORS   5    2125
CORS   6    2126
CORS   7    2127
CORS   8    2128
CORS   9    2129
CORS  10    2130
CORS  11    2131
CORS  12    2132
CORS  13    2133
CORS  14    2134
CORS  15    2135
CORS  16    2136
CORS  17    2137
CORS  18    2138
CORS  19    2139
CORS  20    2140
CORS  21    2141
CORS  22    2142
CORS  23    2143
CORS  24    2144
```

APPENDIX B
LISTING OF THE MACHINE-INDEPENDENT VERSION OF BANDIT

```
C                                                                       BANDIT 5          1
C                         B A N D I T                                   BANDIT 6          2
C                                                                       BANDIT 7          3
C MAIN PROGRAM FOR THE RENUMBERING OF NASTRAN GRID POINTS FOR           BANDIT 8          4
C      REDUCED BANDWIDTH.                                               BANDIT 9          5
C THE NASTRAN DATA DECK MUST CONTAIN A BEGIN BULK CARD IN ITS           BANDIT10          6
C      PROPER PLACE AND TERMINATE WITH AN ENDDATA CARD.                 BANDIT11          7
      DIMENSION A(20)                                                   BANDIT12          8
      COMMON KOM(25000)                                                 JJ     1          9
      COMMON /S/ NN,MM,IH,IB                                            BANDIT14         10
      COMMON /P/ IHO,IHE                                                BANDIT15         11
      COMMON /A/ MAXGRD,MAXDEG,KMOD,NMPC                                BANDIT16         12
      COMMON /B/ IPARAM(20),IARG(5)                                     BANDIT17         13
      COMMON /C/ IWARN,LINE,KORIG,KNEW                                  BANDIT18         14
      COMMON /K/ II(7),KORE,IFL                                         BANDIT19         15
      COMMON /BITS/ NBITIN,NBITEX,IPASS                                 BANDIT20         16
      COMMON /TIME/ STIME,NCOMP                                         BANDIT21         17
      COMMON /NEL/ NEL,TIM2                                             BANDIT22         18
      COMMON /JUL/ ISTART(100),IGNORE(100),IFIRST(100)                  BANDIT23         19
      COMMON /JULL/ IDIM,ISTA,IIG,IFIR,IGDEG,ISCH                       BANDIT24         20
      COMMON /ZERO/ KT                                                  BANDIT25         21
      COMMON /NG/ NGRID,CLOCK                                           BANDIT26         22
      INTEGER EOF                                                       BANDIT27         23
      DATA BEGI,ENDD,SEQG/4HBEGI,4HENDD,4HSEQG/                         BANDIT28         24
C SET NGRID DEFAULT.                                                    BANDIT35         25
      KORE=25000                                                        JJ     2         26
      NGRID = KORE/28                                                   JJ     3         27
C SET SCHEME DEFAULTS.                                                  BANDIT38         28
      IARG(1)=30                                                        BANDIT39         29
      IARG(2)=1                                                         BANDIT40         30
      IARG(3)=2                                                         BANDIT41         31
      IARG(4)=2                                                         BANDIT42         32
      IARG(5)=0                                                         BANDIT43         33
C SET NUMBER OF BITS PER WORD FOR INTERNAL AND EXTERNAL                 BANDIT44         34
C          GRID NUMBERS.                                                BANDIT45         35
      NBITIN=12                                                         BANDIT46         36
      NBITEX=60                                                         BANDIT47         37
    7 FORMAT(1H1,16(/),                                                 BANDIT48         38
     1 36X,57HBBBBBBB     AAAAA    N     N   DDDDDD    IIIIIII   TTTTTTT/BANDIT49         39
     2 36X,57HB     B    A   A    NN    N   D     D      I         T   /BANDIT50         40
     3 36X,57HB     B   A     A   N N   N   D     D      I         T   /BANDIT51         41
     4 36X,57HBBBBBBB   A     A   N  N  N   D     D      I         T   /BANDIT52         42
     5 36X,57HB     B   AAAAAAA   N   N N   D     D      I         T   /BANDIT53         43
     6 36X,57HB     B   A     A   N    NN   D     D      I         T   /BANDIT54         44
     7 36X,57HBBBBBBB   A     A   N     N   DDDDDD    IIIIIII     T   )BANDIT55         45
    8 FORMAT(22(/),48X,34HTHEORY OF STRUCTURES BRANCH - 1844 /          JJ     4         46
     1      48X,38HCOMPUTATION AND MATHEMATICS DEPARTMENT  /            BANDIT57         47
     2      44X,42HNAVAL SHIP RESEARCH AND DEVELOPMENT CENTER /         BANDIT58         48
     3  53X,24HBETHESDA, MARYLAND 20034 )                              BANDIT59         49
    9 FORMAT(/37X,                                                      JJ     5         50
     *  16HREV. 10 MAR 1972 )                                          BANDIT61         51
   10 FORMAT(20A4)                                                      BANDIT62         52
   11 FORMAT(1H ,20A4)                                                  BANDIT63         53
   12 FORMAT(1H1)                                                       BANDIT64         54
   13 FORMAT(///26H TOTAL CP TIME IN BANDIT   ,F9.3,6H  SEC.)           JJ     6         55
      LINE=55                                                           BANDIT66         56
      KNEW=0                                                            BANDIT67         57
      REWIND 8                                                          BANDIT68         58
C PRINT TITLE PAGE.                                                     BANDIT69         59
      WRITE(6,7)                                                        BANDIT70         60
      WRITE(6,8)                                                        BANDIT71         61
      WRITE(6,9)                                                        BANDIT72         62
C INITIALIZE VARIABLES.                                                 BANDIT73         63
      DO 15 J=1,20                                                      BANDIT74         64
   15 IPARAM(J)=0                                                       BANDIT75         65
      IPARAM(12)=4                                                      BANDIT76         66
      IDIM=100                                                          BANDIT77         67
      ISTA=0                                                            BANDIT78         68
      IIG=0                                                             BANDIT79         69
      ISCH=0                                                            BANDIT80         70
      IFIR=0                                                            BANDIT81         71
      IGDEG=0                                                           BANDIT82         72
      DO 18 I=1,IDIM                                                    BANDIT83         73
      ISTART(I)=0                                                       BANDIT84         74
      IFIRST(I)=0                                                       BANDIT85         75
   18 IGNORE(I)=0                                                       BANDIT86         76
      IPASS=0                                                           BANDIT87         77
      NN=0                                                              BANDIT88         78
      MM=0                                                              BANDIT89         79
      MAXGRD=0                                                          BANDIT90         80
      MAXDEG=0                                                          BANDIT91         81
      KMOD=0                                                            BANDIT92         82
      KORIG=0                                                           BANDIT93         83
      KNEW=0                                                            BANDIT94         84
      STIME=0.                                                          BANDIT95         85
      NCOMP=0                                                           BANDIT96         86
      NEL=0                                                             BANDIT97         87
      KT=0                                                              BANDIT98         88
      TIM2=0.                                                           BANDIT99         89
      REWIND 9                                                          BANDI100         90
C READ DECK FOR FIRST TIME.                                             BANDI101         91
      CALL GOOGAN(1,2,5,9)                                              BANDI102         92
C SLICE UP CORE ACCORDING TO SUBROUTINE GRID.                           BANDI103         93
      K2=II(1)*II(2)+1                                                  BANDI104         94
      K3=K2+II(3)*2                                                     BANDI105         95
      K4=K3+II(4)                                                       BANDI106         96
      K5=K4+II(5)                                                       BANDI107         97
      K6=K5+II(6)                                                       BANDI108         98
      K7=K6+II(7)                                                       BANDI109         99
C PROCESS DECK.                                                         BANDI110        100
```

```
      CALL NASNUM(KOM(1),II(1),KOM(K2),II(3),KOM(K3),KOM(K4),KOM(K5),    BANDI111      101
     + KOM(K6),KOM(K7),KOM(1),KOM(K2))                                    BANDI112      102
C       ARRAY STARTING AT LOCATION K7 HAS DIMENSION 2*MAXDEG              BANDI113      103
C PROCESS OUTPUT ACCORDING TO OUTPUT REQUESTS.                            BANDI114      104
C CHECK IF CONNECTION CARDS IN DECK.                                      BANDI115      105
      IF(IPARAM(9).EQ.3)GO TO 19                                          BANDI116      106
      REWIND 8                                                            BANDI117      107
      REWIND 9                                                            BANDI118      108
      FLAG=ENDD                                                           BANDI119      109
      J=0                                                                 BANDI120      110
      K=9                                                                 BANDI121      111
      GO TO 20                                                            BANDI122      112
   19 REWIND 8                                                            BANDI123      113
      J=0                                                                 BANDI124      114
      K=8                                                                 BANDI125      115
      FLAG=ENDD                                                           BANDI126      116
      IF(IPARAM(5).EQ.4)GO TO 20                                          BANDI127      117
      K=9                                                                 BANDI128      118
      IF(IPARAM(6).EQ.3)FLAG=BEGI                                         BANDI129      119
   20 READ(K,10)A                                                         BANDI130      120
      IF(EOF(K).NE.0)CALL BOMBIT(1)                                       BANDI131      121
      J=J+1                                                               BANDI132      122
      IF(IPARAM(10).EQ.5.AND.A(1).NE.SEQG) J=J-1                          BANDI133      123
      IF(MOD(J,LINE).EQ.1)WRITE(6,12)                                     BANDI134      124
      IF(IPARAM(10).EQ.6) WRITE(6,11) A                                   BANDI135      125
      IF(IPARAM(10).EQ.5.AND.A(1).EQ.SEQG) WRITE(6,11)A                   BANDI136      126
      IF(IPARAM(1).EQ.2)WRITE(7,10)A                                      BANDI137      127
      IF(IPARAM(1).EQ.1.AND.A(1).EQ.SEQG) WRITE(7,10) A                   BANDI138      128
      IF(K.NE.8) WRITE(8,10) A                                            BANDI139      129
      IF(A(1).NE.FLAG)GO TO 20                                            BANDI140      130
      IF(FLAG.EQ.ENDD)GO TO 25                                            BANDI141      131
      FLAG=ENDD                                                           BANDI142      132
      K=5                                                                 BANDI143      133
      GO TO 20                                                            BANDI144      134
   25 CONTINUE                                                            JJ     7      135
      IF(IPARAM(5).EQ.3)GO TO 60                                          BANDI147      136
      IF(IPARAM(7).EQ.4)GO TO 60                                          BANDI148      137
      IF(IPARAM(9).EQ.4)GO TO 60                                          BANDI149      138
C USER SUMMARY.                                                           BANDI150      139
      WRITE(6,50) KORIG,KNEW                                              JJ     8      140
   50 FORMAT(23H1***BANDIT USER SUMMARY /                                 BANDI152      141
     1 8X,25HORIGINAL SEMI-BANDWIDTH   ,I9/                               JJ     9      142
     2 8X,20HNEW SEMI-BANDWIDTH    ,I14)                                  JJ    10      143
      WRITE(6,117) IHO,IHE                                                BANDI156      144
  117 FORMAT(8X,18HORIGINAL PROFILE  ,I16/8X,13HNEW PROFILE   ,I21)       JJ    11      145
      WRITE(6,104) NN                                                     BANDI158      146
      WRITE(6,115) NEL                                                    BANDI159      147
      WRITE(6,112) NCOMP                                                  BANDI160      148
      WRITE(6,107) MM                                                     BANDI161      149
  107 FORMAT(8X,22HMAXIMUM NODAL DEGREE    ,I12)                         JJ    12      150
      WRITE(6,116) KT                                                     BANDI163      151
      I=IPARAM(1)                                                         BANDI164      152
      IF(I.EQ.1) WRITE(6,101)                                             BANDI165      153
      IF(I.EQ.2) WRITE(6,102)                                             BANDI166      154
      IF(I.EQ.3) WRITE(6,103)                                             BANDI167      155
  101 FORMAT(8X,34HPUNCH OUTPUT            SEQGP CARDS )                  BANDI168      156
  102 FORMAT(8X,34HPUNCH OUTPUT              ALL CARDS )                  BANDI169      157
  103 FORMAT(8X,34HPUNCH OUTPUT                  NONE )                   BANDI170      158
      WRITE(6,105) MAXGRD,MAXDEG                                          BANDI172      159
  105 FORMAT(18X,8HMAXGRD  ,I11/18X,8HMAXDEG    ,I11)                    JJ    13      160
  111 FORMAT(8X,14HDATE AND TIME ,2A10)                                   BANDI177      161
C IPASS=NUMBER OF PDUP CALLS.                                             BANDI178      162
  104 FORMAT(8X,23HNUMBER OF GRID POINTS    ,I11)                       JJ    14      163
  113 FORMAT(8X,23HNUMBER OF ELEMENTS    ,I14)                          JJ    15      164
  112 FORMAT(8X,22HNUMBER OF COMPONENTS    ,I12)                        JJ    16      165
  115 FORMAT(8X,23HNO. OF POINTS OF ZERO DEGREE ,I6)                      JJ    17      166
      GO TO 70                                                            BANDI184      167
   60 IF(IPARAM(10).EQ.5) WRITE(6,12)                                     BANDI185      168
   70 CONTINUE                                                            BANDI187      169
      REWIND 8                                                            BANDI188      170
      IF(IPARAM(8).EQ.4) STOP 5                                           BANDI190      171
      STOP                                                                BANDI191      172
      END                                                                 BANDI192      173
      SUBROUTINE NASNUM(IG,III,INV,III3,INT,ICC,ILD,NORIG,IP,JG,JNV)      NASNUM 2      174
      DIMENSION A(8),KG(40),LG(40),LINE(10),J(20),ATEMP(4)                NASNUM 3      175
      DIMENSION IG(III,1),INV(III3,2),JG(1),JNV(1)                        NASNUM 4      176
      DIMENSION INT(1),ICC(1),ILD(1),NORIG(1),IP(1)                       NASNUM 5      177
C IP HAS DIMENSION 2*MAXDEG.   JG AND JNV ARE EQUIV TO IG AND INV.        NASNUM 6      178
      COMMON /3/ NN,MM,IH,IB                                              NASNUM 7      179
      COMMON /4/ MAXGRD,MAXDEG,KMOD,NMPC                                  NASNUM 8      180
      COMMON /8/ IPARAM(20),IARG(5)                                       NASNUM 9      181
      COMMON /G/ IWARN,NLINE,KORIG,KNEW                                   NASNUM10      182
      COMMON /ITS/ NJITIN,NJITEX,IPASS                                    NASNUM11      183
      COMMON /K/ II(7),KORE                                               NASNUM12      184
      COMMON /TIME/ TIM2,NCOMP                                            NASNUM13      185
      COMMON /NEL/ NEL                                                    NASNUM14      186
      COMMON /JOL/ ISTART(100),IGNORE(100),IFIRST(100)                    NASNUM15      187
      COMMON /JOLL/ IDIM,ISTA,IIG,IFIR,IGDEG,ISCH                         NASNUM16      188
C THE VARIABLE LINE DEFINED NEAR CARD NASNUM.300 IS NOT THE               NASNUM17      189
C       SAME AS THE VARIABLE LINE APPEARING IN COMMON                     NASNUM18      190
C            IN OTHER ROUTINES.                                           NASNUM19      191
      DIMENSION TYPE(50),WTYPE(50)                                        NASNUM20      192
      DIMENSION F1A(2),F10A(2),F1B(2),F10B(2)                             NASNUM21      193
      DATA BLGI,ENDD,SEQG/4HBEGI,4HENDD,4HSEQG/                           NASNUM22      194
      DATA TYPE/4HBAR,4HCELA,4HCELA,4HCONR,4HLQDR,4HCQDP,4HCQDA,          NASNUM23      195
     1   4HCQDA,4HCQDA,4HCROD,4HCSHE,4HCTRB,4HCTRI,4HCTRI,4HCTRM,         NASNUM24      196
     2   4HCTRP,4HCTUB,4HCTWI,4HENDD,4HMPC*,4HCDAM,4HCDAM,4HCMAS,         NASNUM25      197
     3   4HCMAS,4HCVIS,4HCDAM,4HCDAM,4HCELA,4HCELA,4HCMAS,4HCMAS,         NASNUM26      198
     4   4HCCON,4HCTOR,4HCTRA,4HCTRI,4HCDAM,4HCONM,4HCHIT,4HCIS3,         NASNUM27      199
     5   4HCIS3,4HCIS2,4HCISH,4HCISH,4HCFLU,4HCFLU,4HCFLU,                NASNUM28      200
```

```
     6   4HCTET,4HCHEX,4HCHEX/                                              NASNUM29   201
      DATA WYPE/4H*    ,4HS1* ,4HS2* ,4HOD* ,4HEM* ,4HLT* ,4HO1* ,          NASNUM30   202
     1   4HJ2* ,4HJ3* ,4H*    ,4HAR* ,4HSC* ,4HA1* ,4HA2* ,4HEM* ,          NASNUM31   203
     2   4HLT* ,4HE*  ,4HST* ,4HATA ,4H     ,4HP1* ,4HP2* ,4HS1* ,          NASNUM32   204
     3   4HS2* ,4HJ*  ,4HP3* ,4HP4* ,4HS3* ,4HS4* ,4HS3* ,4HS4* ,          NASNUM33   205
     4   4HEAX*,4HJRG*,4HPRG*,4HARG*,4HI*   ,4H2*   ,4HRI2*,4HD8* ,          NASNUM34   206
     5   4HD20*,4HD4* ,4HD8* ,4H8*   ,4H16* ,4HID2*,4HID3*,4HID4*,          NASNUM35   207
     6   4HRA* ,4HA1* ,4HA2* /                                              NASNUM36   208
      NTYPE=50                                                              NASNUM37   209
      REWIND 8                                                              NASNUM38   210
      REWIND 9                                                              NASNUM39   211
      NMPC=40                                                               NASNUM40   212
      KMOD=2.*FLOAT(MAXGRD)-2.2715*SQRT(1.131*FLOAT(MAXGRD))                NASNUM41   213
      NEW=0                                                                 NASNUM42   214
      IWARN=0                                                               NASNUM43   215
      NEQ=0                                                                 NASNUM44   216
    2 FORMAT(29H1BANDIT INFORMATION MESSAGE - /                             NASNUM45   217
     +19H      40 GRID POINTS/                                              NASNUM46   218
     +28H      RESEQUENCING SUPPRESSED)                                     NASNUM47   219
    4 FORMAT(    19H ***NEW BANDWIDTH =,I6)                                 NASNUM48   220
    5 FORMAT(33H1THE WRONG CARD FOLLOWS THIS CARD/1X,2A4,1P4E16.7,2A4// NASNUM49   221
     2       40H THE CONTINUATION CARD IS REQUIRED NEXT ,                   NASNUM50   222
     3     36HSINCE BANDIT DOES NOT SORT THE DECK. /                        NASNUM51   223
     4 13H FATAL ERROR. )                                                   NASNUM52   224
    6 FORMAT(1H1)                                                           NASNUM53   225
    7 FORMAT(54H1 ONE OR MORE SEQGP CARDS ALREADY APPEAR IN DATA DECK./ NASNUM54   226
     +  55H  RESEQUENCING CANNOT BE REQUESTED.        FATAL ERROR. )       NASNUM55   227
    8 FORMAT(5HSEQGP,3X,2I8,56X)                                            NASNUM56   228
    9 FORMAT(20A4)                                                          NASNUM57   229
   10 FORMAT(2A4,4F16.0,2A4)                                                NASNUM58   230
   11 FORMAT(14 ,5(I8,I11,7X))                                              NASNUM59   231
   12 FORMAT(5HSEQGP,3X,8I8,8X)                                             NASNUM60   232
   14 FORMAT(////26H ***BANDIT WARNING MESSAGE /                            NASNUM61   233
     1 11X,35HTHE WRONG CARD MAY FOLLOW THIS CARD /                         NASNUM62   234
     2  11X,2A4,1P4E16.7,2A4/                                               NASNUM63   235
     3      11X,47HCHECK INPUT DECK TO BE SURE THAT A CONTINUATION ,        NASNUM64   236
     4  42H CARD IS NEITHER MISSING NOR OUT OF SORT.    )                   NASNUM65   237
   15 FORMAT(26H TOTAL CP TIME IN SCHEME = ,F9.3,6H  SEC. )                 NASNUM66   238
   16 FORMAT(1H1,5(20HINTERNAL    ORIGINAL,6X)/                             NASNUM67   239
     11H ,5(20HGRID NO.    GRID PT.,6X))                                    NASNUM68   240
C RETURN IF RESEQUENCING IS NOT DESIRED.                                    NASNUM69   241
      IF(IPARAM(5).EQ.3)RETURN                                              NASNUM70   242
C CHECK IF SEQGP CARDS ALREADY APPEAR IN DECK.                              NASNUM71   243
      IF(IPARAM(7).EQ.3)GO TO 22                                            NASNUM72   244
C ABORT BANDIT SINCE SEQGP CARDS ALREADY APPEAR IN DECK.                    NASNUM73   245
      WRITE(6,7)                                                            NASNUM74   246
      CALL BOMBIT(3)                                                        NASNUM75   247
C READ AND EXTRACT CONNECTION CARDS FROM DECK.                              NASNUM76   248
   22 CALL SOOGAN(2,1,9,8)                                                  NASNUM77   249
      REWIND 8                                                              NASNUM78   250
      REWIND 9                                                              NASNUM79   251
C INITIALIZE EXPANDABLE CORE.                                               NASNUM80   252
      DO 30 I=1,KORE                                                        NASNUM81   253
   30 JG(I)=0                                                               NASNUM82   254
C READ CARD.                                                                NASNUM83   255
   40 READ(8,10)F1A,(A(I),I=1,4),F10A                                       NASNUM84   256
C DETERMINE CARD TYPE.                                                      NASNUM85   257
   45 ITYPE=0                                                               NASNUM86   258
      DO 50 I=1,NTYPE                                                       NASNUM87   259
   50 IF(F1A(1).EQ.TYPE(I).AND.F1A(2).EQ.WYPE(I)) ITYPE=I                   NASNUM88   260
      IF(ITYPE.EQ.0)GO TO 40                                                NASNUM89   261
      IF(ITYPE.EQ.19)GO TO 500                                              NASNUM90   262
      IF(ITYPE.EQ.20.AND.IPARAM(4).EQ.3)GO TO 40                            NASNUM91   263
C READ CONTINUATION TO CARD JUST READ.                                      NASNUM92   264
      READ(8,10)F1B,(A(I),I=5,8),F10B                                       NASNUM93   265
C CHECK EACH LOGICAL CARD FOR PROPER SORT.                                  NASNUM94   266
      IF(F1B(1).EQ.F10A(1).AND.F1B(2).EQ.F10A(2)) GO TO 60                  NASNUM95   267
C--- IF FOLLOWING CARD TYPES ARE OUT OF SORT, NO ERROR                      NASNUM96   268
      IF(ITYPE.EQ.1.OR.ITYPE.EQ.4)GO TO 56                                  NASNUM97   269
      IF(ITYPE.EQ.32)GO TO 56                                               NASNUM98   270
      IF(ITYPE.EQ.33)GO TO 56                                               NASNUM99   271
      IF(ITYPE.EQ.35)GO TO 56                                               NASNU100   272
      IF(ITYPE.EQ.36)GO TO 56                                               NASNU101   273
      IF(ITYPE.EQ.37)GO TO 56                                               NASNU102   274
      IF(ITYPE.EQ.45.OR.ITYPE.EQ.46) GO TO 56                               NASNU103   275
C--- IF FOLLOWING CARD TYPES ARE OUT OF SORT, POSSIBLE ERROR (GIVE          NASNU104   276
C---             WARNING MESSAGE)                                           NASNU105   277
      IWARN=IWARN+1                                                         NASNU106   278
      IF(MOD(IWARN,6).EQ.1)WRITE(6,6)                                       NASNU107   279
      IF(ITYPE.EQ. 2) GO TO 54                                              NASNU108   280
      IF(ITYPE.EQ. 3) GO TO 54                                              NASNU109   281
      IF(ITYPE.EQ.10) GO TO 54                                              NASNU110   282
      IF(ITYPE.EQ.17) GO TO 54                                              NASNU111   283
      IF(ITYPE.GE.21.AND.ITYPE.LE.31)GO TO 54                               NASNU112   284
C--- FOR OTHER CARD TYPES OUT OF SORT, ABORT BANDIT                         NASNU113   285
   52 WRITE(6,5)F1A,(A(I),I=1,4),F10A                                       NASNU114   286
      CALL BOMBIT(2)                                                        NASNU115   287
   54 WRITE(6,14)F1A,(A(I),I=1,4),F10A                                      NASNU116   288
C SAVE CONTENTS OF THE SECOND CARD OF THE PAIR.                             NASNU117   289
   56 DO 58 I=1,4                                                           NASNU118   290
      ATEMP(I)=A(I+4)                                                       NASNU119   291
   58 A(I+4)=0.                                                             NASNU120   292
C INITIALIZE KG AND LG.                                                     NASNU121   293
   60 DO 70 I=1,NMPC                                                        NASNU122   294
      KG(I)=0                                                               NASNU123   295
   70 LG(I)=0                                                               NASNU124   296
      LOOP=1                                                                NASNU125   297
      NCON=4                                                                NASNU126   298
C SET UP KG AND LG.  **                                                     NASNU127   299
      GO TO (160,220,220,200,120,120,120,120,120,180,120,140,140,           NASNU128   300
```

```
     1       140,140,140,180,120,500,230,220,220,220,220,180,180,      NASNU129   301
     2       180,180,180,180,180,160,160,110,114,118,118,140, 80,      NASNU130   302
     3        85,120, 90, 80, 95,200,114,110,120, 90, 90),ITYPE        NASNU131   303
C* CIS3D8,CISH8                                                        NASNU132   304
   80 DO 81 I=1,7                                                      NASNU133   305
   81 KG(I)=A(I+1)+0.5                                                 NASNU134   306
      NCON=8                                                           NASNU135   307
      READ(8,10) F1A,A(1),A(2),A(3),A(4),F10A                          NASNU136   308
      IF(F1A(1).NE.F10B(1).OR.F1A(2).NE.F10B(2)) GO TO 100             NASNU137   309
      KG(8)=A(1)+0.5                                                   NASNU138   310
      GO TO 250                                                        NASNU139   311
C* CIS3D20                                                            NASNU140   312
   85 DO 86 I=1,7                                                      NASNU141   313
   86 KG(I)=A(I+1)+0.5                                                 NASNU142   314
      NCON=20                                                          NASNU143   315
      READ(8,10) F1A,A(1),A(2),A(3),A(4),F10A                          NASNU144   316
      IF(F1A(1).NE.F10B(1).OR.F1A(2).NE.F10B(2)) GO TO 100             NASNU145   317
      READ(8,10) F1B,A(5),A(6),A(7),A(8),F10B                          NASNU146   318
      IF(F1B(1).NE.F10A(1).OR.F1B(2).NE.F10A(2)) GO TO 52              NASNU147   319
      DO 87 I=8,15                                                     NASNU148   320
   87 KG(I)=A(I-7)+0.5                                                 NASNU149   321
      READ(8,10) F1A,A(1),A(2),A(3),A(4),F10A                          NASNU150   322
      IF(F1A(1).NE.F10B(1).OR.F1A(2).NE.F10B(2)) GO TO 100             NASNU151   323
      READ(8,10) F1B,A(5),A(6),A(7),A(8),F10B                          NASNU152   324
      IF(F1B(1).NE.F10A(1).OR.F1B(2).NE.F10A(2)) GO TO 52              NASNU153   325
      DO 88 I=16,20                                                    NASNU154   326
   88 KG(I)=A(I-15)+0.5                                                NASNU155   327
      GO TO 250                                                        NASNU156   328
C* CIS2D8,CHEXA1,CHEXA2                                               NASNU157   329
   90 DO 91 I=1,6                                                      NASNU158   330
   91 KG(I)=A(I+2)+0.5                                                 NASNU159   331
      NCON=8                                                           NASNU160   332
      READ(8,10) F1A,A(1),A(2),A(3),A(4),F10A                          NASNU161   333
      IF(F1A(1).NE.F10B(1).OR.F1A(2).NE.F10B(2)) GO TO 100             NASNU162   334
      DO 92 I=7,8                                                      NASNU163   335
   92 KG(I)=A(I-6)+0.5                                                 NASNU164   336
      GO TO 250                                                        NASNU165   337
C* CISH16                                                             NASNU166   338
   95 DO 96 I=1,7                                                      NASNU167   339
   96 KG(I)=A(I+1)+0.5                                                 NASNU168   340
      NCON=16                                                          NASNU169   341
      READ(8,10) F1A,A(1),A(2),A(3),A(4),F10A                          NASNU170   342
      IF(F1A(1).NE.F10B(1).OR.F1A(2).NE.F10B(2)) GO TO 100             NASNU171   343
      READ(8,10) F1B,A(5),A(6),A(7),A(8),F10B                          NASNU172   344
      IF(F1B(1).NE.F10A(1).OR.F1B(2).NE.F10A(2)) GO TO 52              NASNU173   345
      DO 97 I=9,15                                                     NASNU174   346
   97 KG(I)=A(I-7)+0.5                                                 NASNU175   347
      READ(8,10) F1A,A(1),A(2),A(3),A(4),F10A                          NASNU176   348
      IF(F1A(1).NE.F10B(1).OR.F1A(2).NE.F10B(2)) GO TO 100             NASNU177   349
      KG(16)=A(1)+0.5                                                  NASNU178   350
      GO TO 250                                                        NASNU179   351
  100 F1A(1)=F10B(1)                                                   NASNU180   352
      F1A(2)=F10B(2)                                                   NASNU181   353
       DO 101 I=1,4                                                    NASNU182   354
  101 A(I)=A(I+4)                                                      NASNU183   355
      F10A(1)=F10B(1)                                                  NASNU184   356
      F10A(2)=F10B(2)                                                  NASNU185   357
      GO TO 52                                                         NASNU186   358
C* CTRAPRG,CFLUID4                                                    NASNU187   359
  110 DO 112 I=1,4                                                     NASNU188   360
  112 KG(I)=A(I+1)+0.5                                                 NASNU189   361
      GO TO 250                                                        NASNU190   362
C* CTRIARG,CFLUID3                                                    NASNU191   363
  114 DO 116 I=1,3                                                     NASNU192   364
  116 KG(I)=A(I+1)+0.5                                                 NASNU193   365
      GO TO 250                                                        NASNU194   366
C* CONM1, CONM2                                                       NASNU195   367
  118 KG(1)=A(2)+0.5                                                   NASNU196   368
      KG(2)=KG(1)                                                      NASNU197   369
      GO TO 250                                                        NASNU198   370
C* CQDMEM,CQDPLT,CQUAD1,CQUAD2,CQUAD3,CSHEAR,CTWIST,CIS2D4,CTETRA      NASNU199   371
  120 DO 130 I=1,4                                                     NASNU200   372
  130 KG(I)=A(I+2)+0.5                                                 NASNU201   373
      GO TO 250                                                        NASNU202   374
C* CTRBSC, CTRIA1, CTRIA2, CTRMEM, CTRPLT, CHTTRI2                    NASNU203   375
  140 DO 150 I=1,3                                                     NASNU204   376
  150 KG(I)=A(I+2)+0.5                                                 NASNU205   377
      GO TO 250                                                        NASNU206   378
C* CBAR, CCONEAX, CTORDRG                                             NASNU207   379
  160 DO 170 I=1,2                                                     NASNU208   380
  170 KG(I)=A(I+2)+0.5                                                 NASNU209   381
      GO TO 250                                                        NASNU210   382
C* CROD, CTUBE, CVISC, CDAMP3, CDAMP4, CELAS3, CELAS4, CMASS3, CMASS4  NASNU211   383
  180 DO 190 I=1,2                                                     NASNU212   384
      KG(I)=A(I+2)+0.5                                                 NASNU213   385
  190 LG(I)=A(I+6)+0.5                                                 NASNU214   386
C SET LOOP=2 SINCE 2 ELEMENTS MAY BE DEFINED ON ONE CARD.              NASNU215   387
      LOOP=2                                                           NASNU216   388
      GO TO 250                                                        NASNU217   389
C* CONROD,CFLUID2                                                     NASNU218   390
  200 DO 210 I=1,2                                                     NASNU219   391
  210 KG(I)=A(I+1)+0.5                                                 NASNU220   392
      GO TO 250                                                        NASNU221   393
C* CELAS1, CELAS2, CDAMP1, CDAMP2, CMASS1, CMASS2                     NASNU222   394
  220 KG(1)=A(3)+0.5                                                   NASNU223   395
      KG(2)=A(5)+0.5                                                   NASNU224   396
      GO TO 250                                                        NASNU225   397
C PROCESS MPC CARDS.                                                   NASNU226   398
  230 NCON=NMPC                                                        NASNU227   399
      KG(1)=A(2)+0.5                                                   NASNU228   400
```

```
      KG(2)=A(5)+0.5                                                    NASNU229   401
      I=2                                                               NASNU230   402
  240 READ(8,10)F1A,(A(J),J=1,4),F10A                                   NASNU231   403
      IF(F10B(1).NE.F1A(1).OR.F10B(2).NE.F1A(2)) GO TO 250              NASNU232   404
      I=I+1                                                             NASNU233   405
      IF(I.GT.NMPC)GO TO 245                                            NASNU234   406
      F10B(1)=F10A(1)                                                   NASNU235   407
      F10B(2)=F10A(2)                                                   NASNU236   408
      KK=2                                                              NASNU237   409
      IF(MOD(I,2).EQ.0)KK=1                                             NASNU238   410
      KG(I)=A(KK)+0.5                                                   NASNU239   411
      GO TO 240                                                         NASNU240   412
  245 WRITE(6,246) NMPC                                                 NASNU241   413
  246 FORMAT(36H1 AN MPC EQUATION CONTAINS MORE THAN,I5,8H  TERMS./     NASNU242   414
     +  14H  FATAL ERROR. )                                            NASNU243   415
      CALL BOMBIT(5)                                                    NASNU244   416
C PROCESS KG (AND LG IF LOOP=2) ARRAY.                                  NASNU245   417
  250 DO 480 KK=1,LOOP                                                  NASNU246   418
      IF(KK.EQ.1)GO TO 300                                              NASNU247   419
      DO 260 I=1,4                                                      NASNU248   420
  260 KG(I)=LG(I)                                                       NASNU249   421
C SCATTER SEARCH AND CONVERT KG TO TEMPORARY SET OF INTERNAL NUMBERS.    NASNU250   422
  300 CALL SCAT(KG,NCON,NEW,INV,II3,NORIG)                              NASNU251   423
      IF(ITYPE.NE.20)GO TO 420                                          NASNU252   424
C SAVE MPC GRID POINTS FOR LATER PROCESSING BY TIGER.                   NASNU253   425
      NEQ=NEQ+1                                                         NASNU254   426
      WRITE(11)KG                                                       NASNU255   427
      GO TO 45                                                          NASNU256   428
C FILL CONNECTION TABLE ARRAY IG.                                       NASNU257   429
  420 IEND=NCON-1                                                       NASNU258   430
      NEL=NEL+1                                                         NASNU259   431
      DO 450 I=1,IEND                                                   NASNU260   432
      L=I+1                                                             NASNU261   433
      DO 450 J=L,NCON                                                   NASNU262   434
  450  CALL SETIG(KG(I),KG(J),IG,II1,NORIG)                             NASNU263   435
  480 CONTINUE                                                          NASNU264   436
      IF(F1B(1).EQ.F10A(1).AND.F1B(2).EQ.F10A(2)) GO TO 40              NASNU265   437
      IF(NCON.GE.8) GO TO 40                                            NASNU266   438
      F1A(1)=F1B(1)                                                     NASNU267   439
      F1A(2)=F1B(2)                                                     NASNU268   440
      DO 495 I=1,4                                                      NASNU269   441
  495 A(I)=ATEMP(I)                                                     NASNU270   442
      F10A(1)=F10B(1)                                                   NASNU271   443
      F10A(2)=F10B(2)                                                   NASNU272   444
      GO TO 45                                                          NASNU273   445
  500 NN=NEW                                                            NASNU274   446
      IF(NEW.GT.0) GO TO 502                                            NASNU275   447
      WRITE(6,2)                                                        NASNU276   448
      IPARAM(9)=4                                                       NASNU277   449
      RETURN                                                            NASNU278   450
  502 IF(IPARAM(4).EQ.3)GO TO 505                                       NASNU279   451
C MODIFY CONNECTION TABLE TO ACCOUNT FOR MPC EQUATIONS.                 NASNU280   452
      CALL TIGER(NEQ,IG,II1,ILD,NORIG)                                  NASNU281   453
      NDEP=NN                                                           NASNU282   454
      CALL FIXIT(ILD,NDEP)                                              NASNU283   455
C GENERATE NEW IG AND NORIG ARRAYS.                                     NASNU284   456
  505 CALL BRIGIT(IG,II1,INV,II3,INT,ICC,NORIG,IP)                      NASNU285   457
C PRINT INTERNAL/EXTERNAL CORRESPONDENCE TABLE.                         NASNU286   458
      LEN=50                                                            NASNU287   459
      IF(IPARAM(10).EQ.5) GO TO 560                                     NASNU288   460
      J=0                                                               NASNU289   461
  510 WRITE(6,19)                                                       NASNU290   462
  520 J=J+1                                                             NASNU291   463
      KEND=0                                                            NASNU292   464
      DO 530 K=1,9,2                                                    NASNU293   465
      L=J+LEN*(K-1)/2                                                   NASNU294   466
      LINE(K)=L                                                         NASNU295   467
      IF(L.GT.NEW) GO TO 550                                            NASNU296   468
      KEND=K+1                                                          NASNU297   469
  530 LINE(K+1)=NORIG(L)                                                NASNU298   470
  550 CONTINUE                                                          NASNU299   471
      IF(KEND.EQ.0)GO TO 560                                            NASNU300   472
      WRITE(6,11)(LINE(K),K=1,KEND)                                     NASNU301   473
      IF(MOD(J,LEN).NE.0)GO TO 520                                      NASNU302   474
      J=J+4*LEN                                                         NASNU303   475
      IF(J.LT.NEW) GO TO 510                                            NASNU304   476
  560 CONTINUE                                                          NASNU305   477
C CONVERT ISTART,IGNORE,IFIRST FROM ORIGINAL TO INTERNAL NUMBERS.       NASNU306   478
      I=ISTA+IIG+IFIR                                                   NASNU307   479
      IF(I.LE.0) GO TO 570                                              NASNU308   480
      CALL FLIP(ISTART,ISTA,INV,II3,ICC)                                NASNU309   481
      CALL FLIP(IGNORE,IIG ,INV,II3,ICC)                                NASNU310   482
      CALL FLIP(IFIRST,IFIR,INV,II3,ICC)                                NASNU311   483
      IF(IPARAM(10).EQ.5) GO TO 570                                     NASNU312   484
C PRINT INTERNAL NUMBERS FOR $-CARDS.                                   NASNU313   485
      WRITE(6,561)                                                      NASNU314   486
  561 FORMAT(30H1    $ CARDS (INTERNAL NUMBERS) /)                      NASNU315   487
      IF(ISTA.GT.0) WRITE(6,562) (ISTART(I),I=1,ISTA)                   NASNU316   488
      IF(IIG .GT.0) WRITE(6,564) (IGNORE(I),I=1,IIG )                   NASNU317   489
      IF(IFIR.GT.0) WRITE(6,566) (IFIRST(I),I=1,IFIR)                   NASNU318   490
  562 FORMAT(9H $START  ,20I5/100(9X,20I5/))                            NASNU319   491
  564 FORMAT(9H $IGNORE ,20I5/100(9X,20I5/))                            NASNU320   492
  566 FORMAT(9H $FIRST  ,20I5/100(9X,20I5/))                            NASNU321   493
  570  CONTINUE                                                         NASNU322   494
C SET UP LIST OF POINTS TO IGNORE IN INT ARRAY.                         NASNU323   495
      K=0                                                               NASNU324   496
      IF(IPARAM(4).EQ.3) GO TO 920                                      NASNU325   497
      IF(NDEP.LE.0) GO TO 920                                           NASNU326   498
C  MPC DEPENDENT POINTS FIRST.                                          NASNU327   499
      DO 915 I=1,NDEP                                                   NASNU328   500
```

```
      J=ILD(I)                                                          NASNU329   501
      IF(J.LE.0) GO TO 915                                              NASNU330   502
      K=K+1                                                             NASNU331   503
      INT(K)=ICC(J)                                                     NASNU332   504
      IF(K.GE.MAXGRD) CALL FIXIT(INT,K)                                 NASNU333   505
  915 CONTINUE                                                          NASNU334   506
  920 IF(IGDEG.LE.0) GO TO 940                                          NASNU335   507
C  GRID POINTS WITH DEGREE.GT.IGDEG SECOND.                             NASNU336   508
      IF(IGDEG.GE.MM) GO TO 940                                         NASNU337   509
      CALL DEGREE(IG,II1,INV)                                           NASNU338   510
C HERE, INV(I)=DEGREE OF GRID POINT I                                   NASNU339   511
      DO 930 I=1,NN                                                     NASNU340   512
      IF(JNV(I).LE.IGDEG) GO TO 930                                     NASNU341   513
      K=K+1                                                             NASNU342   514
      INT(K)=I                                                          NASNU343   515
      IF(K.GE.MAXGRD) CALL FIXIT(INT,K)                                 NASNU344   516
  930 CONTINUE                                                          NASNU345   517
  940 IF(IIG.LE.0) GO TO 960                                            NASNU346   518
C  $IGNORE POINTS THIRD.                                                NASNU347   519
      DO 950 I=1,IIG                                                    NASNU348   520
      J=IGNORE(I)                                                       NASNU349   521
      IF(J.LE.0) GO TO 950                                              NASNU350   522
      K=K+1                                                             NASNU351   523
      INT(K)=J                                                          NASNU352   524
      IF(K.GE.MAXGRD) CALL FIXIT(INT,K)                                 NASNU353   525
  950 CONTINUE                                                          NASNU354   526
C K=NUMBER OF POINTS TO BE IGNORED BEFORE COMPRESSING LIST.             NASNU355   527
  960 IF(K.LE.0) GO TO 970                                              NASNU356   528
C DELETE POINTS LISTED IN INT ARRAY FROM CONNECTION TABLE IG.           NASNU357   529
      CALL MORRIS(INT,K,IG,II1)                                         NASNU358   530
  970 CONTINUE                                                          NASNU359   531
C RENUMBER NODES WITH SUBROUTINE SCHEME.                                NASNU360   532
      IF(IPARAM(10).EQ.6) IARG(5)=1                                     NASNU361   533
      II8=II3/2                                                         NASNU362   534
      CALL SCHEME(IARG(1),IARG(2),IARG(3),IARG(4),IARG(5),IG,II1,       NASNU364   535
     + JNV(1),JNV(II8+1),JNV(2*II8+1),JNV(3*II8+1),INT,ICC,ILD,IP)      NASNU365   536
      IF(IPARAM(10).EQ.5) GO TO 580                                     NASNU368   537
      WRITE(6,4)IB                                                      NASNU370   538
C WRITE NEW NASTRAN DATA DECK.                                          NASNU371   539
  580 READ(9,9)B                                                        NASNU372   540
      WRITE(8,9)B                                                       NASNU373   541
      IF(B(1).NE.BEGI)GO TO 580                                         NASNU374   542
  590 READ(9,9)B                                                        NASNU375   543
      IF(B(1).GE.SEQG.OR.B(1).EQ.ENDD)GO TO 600                         NASNU376   544
      WRITE(8,9)B                                                       NASNU377   545
      GO TO 590                                                         NASNU378   546
C WRITE SEQGP CARDS.                                                    NASNU379   547
  600 KREM=MOD(NEW,4)                                                   NASNU380   548
      IF(NEW.GE.4) GO TO 605                                            NASNU381   549
      KBEG=1                                                            NASNU382   550
      GO TO 612                                                         NASNU383   551
605   IEND=NEW-KREM-3                                                   NASNU384   552
      DO 610 K=1,IEND,4                                                 NASNU385   553
      L=K+3                                                             NASNU386   554
  610 WRITE(8,12) (NORIG(I),ILD(I),I=K,L)                               NASNU387   555
      IF(KREM.EQ.0)GO TO 620                                            NASNU388   556
      KBEG=IEND+4                                                       NASNU389   557
  612 DO 615 I=KBEG,NEW                                                 NASNU390   558
  615 WRITE(8,8) NORIG(I),ILD(I)                                        NASNU391   559
C WRITE THE REMAINDER OF THE NASTRAN DECK.                              NASNU392   560
  620 WRITE(8,9)B                                                       NASNU393   561
      IF(B(1).EQ.ENDD)GO TO 700                                         NASNU394   562
      READ(9,9)B                                                        NASNU395   563
      GO TO 620                                                         NASNU396   564
  700 CONTINUE                                                          NASNU397   565
      IF(IPARAM(10).EQ.5) GO TO 900                                     NASNU398   566
C PRINT ORIGINAL GRID POINT CONNECTION TABLE.                           NASNU399   567
      MAXD=MM                                                           NASNU400   568
      L=MAXD/11+1                                                       NASNU401   569
      L=LEN/L                                                           NASNU402   570
705   FORMAT(10H1     GRID,5X,5H  MAX,15X,13H*CONNECTIONS*,5X,          NASNU403   571
     +      23H(ORIGINAL GRID NUMBERS) /5X,                            NASNU404   572
     +      20HPOINT COMP DIST DEGR ,11(8X,1H*) )                      NASNU405   573
710   FORMAT(I10,3I5,11I9/25(25X,11I9/))                                NASNU406   574
      DO 750 I=1,NN                                                     NASNU407   575
      IF(MOD(I,L).EQ.1) WRITE(6,705)                                    NASNU408   576
      DO 720 J=1,MAXD                                                   NASNU409   577
720   IP(J)=0                                                           NASNU410   578
C CALCULATE MDIST AND PRINT TABLE.                                      NASNU411   579
      MDIST=0                                                           NASNU412   580
      DO 725 J=1,MAXD                                                   NASNU413   581
      K=IG(I,J)                                                         JJ    18   582
      IF(K.EQ.0) GO TO 725                                              NASNU415   583
      MDIST=MAXO(MDIST,IABS(I-K))                                       NASNU416   584
      IP(J)=NORIG(K)                                                    NASNU417   585
725   CONTINUE                                                          NASNU418   586
      K=NORIG(I)                                                        NASNU419   587
      IP1=INV(I,1)                                                      NASNU420   588
      IP2=INV(MAXGRD+I,1)                                               NASNU421   589
750   WRITE(6,710) K,IP1,MDIST,IP2,(IP(J),J=1,MAXD)                     NASNU422   590
C PRINT CONNECTION TABLE FOR RENUMBERED NUMBERS.                        NASNU423   591
      DO 780 I=1,NEW                                                    NASNU424   592
  780 ICC(I)=ILD(I)                                                     NASNU425   593
      CALL SWITCH(IG,II1,INT,ICC,IP(1),IP(MAXDEG+1))                    NASNU426   594
      CALL DEGREE(IG,II1,JNV(II8+1))                                    NASNU427   595
      L=COMPNT(IG,II1,JNV(1),JNV(II8+1),JNV(3*II8+1),ICC)               NASNU428   596
      L=MAXD/26+1                                                       NASNU429   597
      L=LEN/L                                                           NASNU430   598
805   FORMAT(37H1LABEL  COMP MDIST  DEGR  CONNECTIONS ,10X,             NASNU431   599
     +       20H(RENUMBERED NUMBERS) )                                  NASNU432   600
```

```
  810 FORMAT(5I6,20I3/ 25(25X,21I5/))                                   NASNU433    601
      DO 850 I=1,NN                                                     NASNU434    602
      IF(MOD(I,L).EQ.1) WRITE(6,805)                                    NASNU435    603
      DO 820 J=1,MAXD                                                   NASNU436    604
  820 IP(J)=0                                                           NASNU437    605
C CALCULATE MDIST AND PRINT TABLE.                                      NASNU438    606
      MDIST=0                                                           NASNU439    607
      DO 825 J=1,MAXD                                                   NASNU440    608
      K=IG(I,J)                                                         JJ    19    609
      IF(K.EQ.J) GO TO 825                                              NASNU442    610
      MDIST=MAX0(MDIST,IABS(I-K))                                       NASNU443    611
      IP(J)=K                                                           NASNU444    612
  825 CONTINUE                                                          NASNU445    613
C INV(I,1)=ICC(I)  BEFORE PACKING                                       NASNU446    614
C INV(MAXGRD+I,1)=IDEG(I)  BEFORE PACKING                               NASNU447    615
      IP1=INV(I,1)                                                      NASNU448    616
      IP2=INV(MAXGRD+I,1)                                               NASNU449    617
  850 WRITE(6,810) I,IP1,MDIST,IP2,(IP(J),J=1,MAXD)                     NASNU450    618
  900 RETURN                                                            NASNU451    619
      END                                                               NASNU452    620
      SUBROUTINE FLIP(LIST,N,INV,II3,ICC)                               FLIP   2    621
C CONVERT S-ARRAY LIST OF LENGTH N FROM ORIGINAL TO INTERNAL NUMBERS.   FLIP   3    622
      COMMON /A/ MAXGRD,MAXDEG,KMOD                                     FLIP   4    623
      DIMENSION LIST(1),INV(II3,2),ICC(1)                               FLIP   5    624
C CHECK FOR DUPLICATE AND ZERO ENTRIES AND REDUCE N IF NECESSARY.       FLIP   6    625
      CALL FIXIT(LIST,N)                                                FLIP   7    626
      IF(N.LE.0) RETURN                                                 FLIP   8    627
      DO 20 I=1,N                                                       FLIP   9    628
      J=LIST(I)                                                         FLIP  10    629
      IF(J.LE.0) GO TO 30                                               FLIP  11    630
      LOC=J-1                                                           FLIP  12    631
   10 LOC=MOD(LOC,KMOD)+1                                               FLIP  13    632
      IF(INV(LOC,1).EQ.0) GO TO 30                                      FLIP  14    633
      IF(INV(LOC,1).NE.J) GO TO 10                                      FLIP  15    634
      K=INV(LOC,2)                                                      FLIP  16    635
      LIST(I)=ICC(K)                                                    FLIP  17    636
   20 CONTINUE                                                          FLIP  18    637
      RETURN                                                            FLIP  19    638
C ABORT BANDIT DUE TO ILLEGAL GRID POINT REFERENCE ON $-CONTROL CARD.    FLIP  20    639
   30 WRITE(6,40) J                                                     FLIP  21    640
   40 FORMAT(11H1GRID POINT ,I10,30H  APPEARING ON A $ CARD IS NOT  ,   FLIP  22    641
     + 25H A STRUCTURAL GRID POINT.  /13H FATAL ERROR.  )               FLIP  23    642
      CALL BOMBIT(8)                                                    FLIP  24    643
      END                                                               FLIP  25    644
      SUBROUTINE GOOGAN(KA,KB,NIN,NOUT)                                 GOOGAN 2    645
C THIS ROUTINE READS A NASTRAN DATA DECK AND RIGHT-ADJUSTS ALL          GOOGAN 3    646
C       BULK DATA IN ITS FIELDS.                                        GOOGAN 4    647
C IN ADDITION, THE CALLING ARGUMENTS PROVIDE THE FOLLOWING OPTIONS -    GOOGAN 5    648
C       KA=1, PROCESS ALL CARDS IN THE NASTRAN DATA DECK, OR            GOOGAN 6    649
C         =2, PROCESS ONLY THOSE CARDS WITH A C OR G IN COLUMN 1,       GOOGAN 7    650
C             MPC CARDS, AND THOSE CONTINUATION CARDS WITH ALL          GOOGAN 8    651
C             NUMERIC FIELDS.  THE ENDDATA CARD IS WRITTEN IN ANY CASE. GOOGAN 9    652
C       KB = 1, CONVERT ALL 8 COLUMN FIELDS TO 16 COLUMN FIELDS, OR     GOOGAN10    653
C          = 2, THE FIELD WIDTHS REMAIN UNCHANGED.                      GOOGAN11    654
C       NIN = THE LOGICAL UNIT FROM WHICH THE INPUT DECK IS READ.       GOOGAN12    655
C       NOUT = THE LOGICAL UNIT ON WHICH THE OUTPUT IS WRITTEN.         GOOGAN13    656
C NEITHER NIN NOR NOUT ARE REWOUND IN THIS ROUTINE.                     GOOGAN14    657
C IF AN ASTERISK APPEARS IN FIELD 1 AFTER THE MNEMONIC, IT IS LEFT-     GOOGAN15    658
C       ADJUSTED AGAINST THE MNEMONIC.                                  GOOGAN16    659
C THE FOLLOWING TWO (2) CARDS ARE REQUIRED IN THE DATA DECK -           GOOGAN17    660
C       (1) A BEGIN BULK CARD TO INDICATE THE BEGINNING OF THE          GOOGAN18    661
C           BULK DATA DECK, AND                                         GOOGAN19    662
C       (2) AN ENDDATA CARD TO INDICATE THE END OF THE DATA DECK.       GOOGAN20    663
C ALL CARDS PRECEDING THE BEGIN BULK CARD ARE WRITTEN ON NOUT IFF KA=1. GOOGAN21    664
      DIMENSION ANUM(10)                                                GOOGAN22    665
      COMMON A(80),IP(40)                                               GOOGAN23    666
      COMMON IA,IB,ICARD,IFLAG,J,JNB,L,MKHOLD,MKINSR,MKNIN              GOOGAN24    667
      COMMON NBLANK,NFIELD,I,ICOL,IFIELD,IPROC,ITYPE                    GOOGAN25    668
      COMMON K,KAST,KBLK,MKI,MKJ,NCOL,NIP,NN                            GOOGAN26    669
      COMMON /A/ MAXGRD,MAXDEG,KMOD,NMPC                                GOOGAN27    670
      COMMON /B/ IPARAM(20),IARG(5)                                     GOOGAN28    671
      COMMON /DOL/ ISTART(100),IGNORE(100),IFIRST(100)                  GOOGAN29    672
      COMMON /DOLL/ IDIM,ISTA,IIG,IFIR,IGOEG,ISCH                       GOOGAN30    673
      COMMON /NG/ NGRID                                                 GOOGAN31    674
      REAL M,N,II,LL,JJ,KK                                              GOOGAN32    675
      INTEGER EOF                                                       GOOGAN33    676
C DATA CARDS FOR ALPHABET (ALLOWS FOR FUTURE ADDITIONS TO               GOOGAN34    677
C           USER OPTION LIST).                                          GOOGAN35    678
      DATA B,C,D,E,J,M,N,P/1HB,1HC,1HD,1HE,1HG,1HM,1HN,1HP/             GOOGAN36    679
      DATA AA,II,LL,O,R/1HA,1HI,1HL,1HO,1HR/                            GOOGAN37    680
      DATA S,T,U,Y/1HS,1HT,1HU,1HY/                                     GOOGAN38    681
      DATA F,H,JJ,KK/1HF,1HH,1HJ,1HK/                                   GOOGAN39    682
      DATA V,W,X,Z/1HV,1HW,1HX,1HZ/                                     GOOGAN40    683
      DATA ASTER,PLUS,BLANK,DOLLAR/1H*,1H+,1H ,1H$/                     GOOGAN41    684
      DATA ANUM/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/                GOOGAN42    685
      DATA LFLAG/0/                                                     GOOGAN43    686
      DATA IBOMB,IBOM/0,0/                                              GOOGAN44    687
      LFLAG=LFLAG+1                                                     GOOGAN45    688
    9 FORMAT(1H1)                                                       GOOGAN46    689
   10 FORMAT(80A1)                                                      GOOGAN47    690
   11 FORMAT(8A1,4(8X,8A1),3H*XZ,I5/3H*XZ,I5,4(8X,8A1),8A1)             GOOGAN48    691
      ICARD=0                                                           GOOGAN49    692
      MKINSR=12                                                         GOOGAN50    693
      MKNIN=NIN                                                         GOOGAN51    694
C READ EXECUTIVE OR CASE CONTROL CARD.                                  GOOGAN52    695
   20 READ(NIN,10)A                                                     GOOGAN53    696
      IF(EOF(NIN).EQ.0)GO TO 21                                         GOOGAN54    697
      IF(NIN.EQ.MKNIN)CALL BOMBIT(1)                                    GOOGAN55    698
      MKNIN=NIN                                                         GOOGAN56    699
      NIN=MKINSR                                                        GOOGAN57    700
```

```
      MKINSR=MKHOLD                                                     GOOGAN58  701
      GO TO 20                                                          GOOGAN59  702
   21 IFLAG=0                                                           GOOGAN60  703
      ICARD=ICARD+1                                                     GOOGAN61  704
C PROCESS OUTPUT OPTION CARD, IF PRESENT.                               GOOGAN62  705
      IF(A(1).NE.DOLLAR)GO TO 26                                        GOOGAN63  706
      IF(LFLAG.GT.1) GO TO 29                                           GOOGAN64  707
C LOOK FOR FIRST KEYWORD.                                               GOOGAN65  708
      ITYPE=0                                                           GOOGAN66  709
      IF(A(2).EQ.P.AND.A(3).EQ.U)ITYPE=1                                GOOGAN67  710
      IF(A(2).EQ.M.AND.A(3).EQ.P)ITYPE=4                                GOOGAN68  711
      IF(A(2).EQ.S.AND.A(3).EQ.E)ITYPE=5                                GOOGAN69  712
      IF(A(2).EQ.K.AND.A(3).EQ.II)ITYPE=6                               GOOGAN70  713
      IF(A(2).EQ.N.AND.A(3).EQ.AA)ITYPE=8                               GOOGAN71  714
      IF(A(2).EQ.P.AND.A(3).EQ.R)ITYPE=10                               GOOGAN72  715
      IF(A(2).EQ.S.AND.A(3).EQ.C)GO TO 1100                             GOOGAN73  716
      IF(A(2).EQ.S.AND.A(3).EQ.T) GO TO 1200                            GOOGAN74  717
      IF(A(2).EQ.D.AND.A(3).EQ.E) GO TO 1250                            GOOGAN75  718
      IF(A(2).EQ.F.AND.A(3).EQ.II) GO TO 1300                           GOOGAN76  719
      IF(A(2).EQ.II.AND.A(3).EQ.G) GO TO 1350                           GOOGAN77  720
      IF(A(2).EQ.G.AND.A(3).EQ.R) GO TO 1380                            GOOGAN78  721
      IF(A(2).EQ.AA.AND.A(3).EQ.T) ITYPE=12                             GOOGAN79  722
      IF(A(2).EQ.M.AND.A(3).EQ.AA) ITYPE=13                             GOOGAN80  723
      IF(A(2).NE.II.OR.A(3).NE.N)GO TO 1025                             GOOGAN81  724
C     INSERT CARDS FROM ALTERNATE FILE                                  GOOGAN82  725
      IPARAM(11)=1                                                      GOOGAN83  726
      MKHOLD=NIN                                                        GOOGAN84  727
      NIN=MKINSR                                                        GOOGAN85  728
      MKINSR=MKHOLD                                                     GOOGAN86  729
      DO 1021 MKI=2,80                                                  GOOGAN87  730
      MKJ=81-MKI                                                        GOOGAN88  731
 1021 A(MKJ+1)=A(MKJ)                                                   GOOGAN89  732
      IPARAM(6)=4                                                       GOOGAN90  733
 1025 IF(ITYPE.EQ.0)GO TO 26                                            GOOGAN91  734
C LOOK FOR SECOND KEYWORD.                                              GOOGAN92  735
      I=3                                                               GOOGAN93  736
   22 I=I+1                                                             GOOGAN94  737
      IF(I.GE.79)GO TO 26                                               GOOGAN95  738
      IF(A(I).NE.BLANK)GO TO 22                                         GOOGAN96  739
   24 I=I+1                                                             GOOGAN97  740
      IF(I.GE.80)GO TO 26                                               GOOGAN98  741
      IF(A(I).EQ.BLANK)GO TO 24                                         GOOGAN99  742
      J=0                                                               GOOGA100  743
      IF(A(I).EQ.S.AND.A(I+1).EQ.E)J=1                                  GOOGA101  744
      IF(A(I).EQ.AA.AND.A(I+1).EQ.LL)J=2                                GOOGA102  745
      IF(A(I).EQ.N.AND.A(I+1).EQ.O)J=3                                  GOOGA103  746
      IF(A(I).EQ.Y.AND.A(I+1).EQ.E)J=4                                  GOOGA104  747
      IF(A(I).EQ.M.AND.A(I+1).EQ.II) J=5                                GOOGA105  748
      IF(A(I).EQ.M.AND.A(I+1).EQ.AA) J=6                                GOOGA106  749
      IF(J.EQ.0)GO TO 26                                                GOOGA107  750
C SET PARAMETER.                                                        GOOGA108  751
      IPARAM(ITYPE)=J                                                   GOOGA109  752
      GO TO 26                                                          GOOGA110  753
C READ $SCHEME CARD.                                                    GOOGA111  754
 1100 CALL READIT(A,IP,NIP)                                             GOOGA112  755
      ISCH=1                                                            GOOGA113  756
      I=MIN0(NIP,5)                                                     GOOGA114  757
      IF(I.EQ.0) GO TO 29                                               GOOGA115  758
      DO 1110 J=1,I                                                     GOOGA116  759
 1110 IARG(J)=IP(J)                                                     GOOGA117  760
      GO TO 29                                                          GOOGA118  761
C READ $START CARD.                                                     GOOGA119  762
 1200 CALL READIT(A,IP,NIP)                                             GOOGA120  763
      I=ISTA                                                            GOOGA121  764
      ISTA=ISTA+NIP                                                     GOOGA122  765
      IF(ISTA.LE.IDIM) GO TO 1205                                       GOOGA123  766
      IBOM=2                                                            GOOGA124  767
      ISTA=IDIM                                                         GOOGA125  768
      GO TO 29                                                          GOOGA126  769
 1205 DO 1210 J=1,NIP                                                   GOOGA127  770
 1210 ISTART(I+J)=IP(J)                                                 GOOGA128  771
      GO TO 29                                                          GOOGA129  772
C READ $DEGREE CARD.                                                    GOOGA130  773
 1250 CALL READIT(A,IP,NIP)                                             GOOGA131  774
      IGDEG=IP(1)                                                       GOOGA132  775
      GO TO 29                                                          GOOGA133  776
C READ $FIRST CARD.                                                     GOOGA134  777
 1300 CALL READIT(A,IP,NIP)                                             GOOGA135  778
      I=IFIR                                                            GOOGA136  779
      IFIR=IFIR+NIP                                                     GOOGA137  780
      IF(IFIR.LE.IDIM) GO TO 1308                                       GOOGA138  781
      IBOM=2                                                            GOOGA139  782
      IFIR=IDIM                                                         GOOGA140  783
      GO TO 29                                                          GOOGA141  784
 1308 DO 1310 J=1,NIP                                                   GOOGA142  785
 1310 IFIRST(I+J)=IP(J)                                                 GOOGA143  786
      GO TO 29                                                          GOOGA144  787
C READ $IGNORE CARD.                                                    GOOGA145  788
 1350 CALL READIT(A,IP,NIP)                                             GOOGA146  789
      I=IIG                                                             GOOGA147  790
      IIG=IIG+NIP                                                       GOOGA148  791
      IF(IIG.LE.IDIM) GO TO 1360                                        GOOGA149  792
      IBOM=2                                                            GOOGA150  793
      IIG=IDIM                                                          GOOGA151  794
      GO TO 29                                                          GOOGA152  795
 1360 DO 1365 J=1,NIP                                                   GOOGA153  796
 1365 IGNORE(I+J)=IP(J)                                                 GOOGA154  797
      GO TO 29                                                          GOOGA155  798
C READ $GRID CARD.                                                      GOOGA156  799
1380  CALL READIT(A,IP,NIP)                                             GOOGA157  800
```

```
      NGRID=IP(1)                                                    GOOGA158   801
      GO TO 29                                                       GOOGA159   802
C LOOK FOR BEGIN BULK CARD.                                          GOOGA160   803
   26 I=0                                                            GOOGA161   804
   27 I=I+1                                                          GOOGA162   805
      IF(I.GT.75)GO TO 29                                            GOOGA163   806
      IF(A(I).EQ.BLANK)GO TO 27                                      GOOGA164   807
      IF(A(I).NE.B)GO TO 29                                          GOOGA165   808
      IF(A(I+1).NE.E)GO TO 29                                        GOOGA166   809
      IF(A(I+2).NE.G)GO TO 29                                        GOOGA167   810
      IF(A(I+3).NE.II)GO TO 29                                       GOOGA168   811
      IFLAG=1                                                        GOOGA169   812
C LEFT-ADJUST BEGIN BULK CARD.                                       GOOGA170   813
      K=73-I                                                         GOOGA171   814
      DO 28 J=1,72                                                   GOOGA172   815
      IF(J.LE.K)A(J)=A(J+I-1)                                        GOOGA173   816
   28 IF(J.GT.K)A(J)=BLANK                                           GOOGA174   817
      IF(LFLAG.GT.1) GO TO 29                                        GOOGA175   818
C REJECT ILLEGAL PARAMETERS AND SET TO DEFAULTS.                     GOOGA176   819
      IF(IPARAM(1).NE.2.AND.IPARAM(1).NE.3) IPARAM(1)=1              GOOGA177   820
      DO 1450 I=2,9                                                  GOOGA178   821
 1450 IF(IPARAM(I).NE.4) IPARAM(I)=3                                 GOOGA179   822
      IF(IPARAM(10).NE.6) IPARAM(10)=5                               GOOGA180   823
      IF(IPARAM(12).NE.3) IPARAM(12)=4                               GOOGA181   824
      IF(IPARAM(13).NE.4) IPARAM(13)=3                               GOOGA182   825
      CALL GRID(NGRID)                                               GOOGA183   826
      I=ISTA+IIG+IFIR+ISCH+IGDEG                                     GOOGA184   827
      IF(I.LE.0) GO TO 29                                            GOOGA185   828
C CHECK FOR ILLEGAL SCHEME ARGUMENTS.                                GOOGA186   829
      DO 1460 I=1,3                                                  GOOGA187   830
1460  IF(IARG(I).LT.1.OR.IARG(I).GT.MAXGRD) IBOMB=1                  GOOGA188   831
      IF(IARG(4).LT.2.OR.IARG(4).GT.3) IBOMB=1                       GOOGA189   832
      IF(IARG(5).LT.0.OR.IARG(5).GT.1) IBOMB=1                       GOOGA190   833
      WRITE(6,9)                                                     GOOGA191   834
      IF(ISCH.GT.0) WRITE(6,1500) (IARG(I),I=1,5)                    GOOGA192   835
 1500 FORMAT(//,9H $SCHEME ,10I10/200(9X,10I10/))                    GOOGA193   836
      IF(ISTA.GT.0) WRITE(6,1505) (ISTART(I),I=1,ISTA)               GOOGA194   837
 1505 FORMAT(//,9H $START  ,10I10/200(9X,10I10/))                    GOOGA195   838
      IF(IGDEG.GT.0) WRITE(6,1510) IGDEG                             GOOGA196   839
 1510 FORMAT(//,9H $DEGREE ,10I10/200(9X,10I10/))                    GOOGA197   840
      IF(IFIR.GT.0) WRITE(6,1515) (IFIRST(I),I=1,IFIR)               GOOGA198   841
 1515 FORMAT(//,9H $FIRST  ,10I10/200(9X,10I10/))                    GOOGA199   842
      IF(IIG.GT.0) WRITE(6,1520) (IGNORE(I),I=1,IIG)                 GOOGA200   843
 1520 FORMAT(//,9H $IGNORE ,10I10/200(9X,10I10/))                    GOOGA201   844
      IF(IBOMB.EQ.1) CALL BOMBIT(4)                                  GOOGA202   845
      IF(IBOM.EQ.2) CALL BOMBIT(9)                                   GOOGA203   846
   29 IF(KA.EQ.1)WRITE(NOUT,10)A                                     GOOGA204   847
      IF(IFLAG.EQ.1)GO TO 20                                         GOOGA205   848
C RETURN IF RIGHT-ADJUSTING OF CARDS IS NOT NEEDED.                  GOOGA206   849
      IF(IPARAM(5).EQ.3.AND.IPARAM(6).EQ.3)RETURN                    GOOGA207   850
C READ BULK DATA CARD.                                               GOOGA208   851
   30 READ(NIN,10)A                                                  GOOGA209   852
      IF(EOF(NIN).EQ.0)GO TO 31                                      GOOGA210   853
      IF(NIN.EQ.MKNIN) CALL BOMBIT(1)                                GOOGA211   854
C     SWITCH INPUT FILES                                             GOOGA212   855
      MKHOLD=NIN                                                     GOOGA213   856
      NIN=MKINSR                                                     GOOGA214   857
      MKINSR=MKHOLD                                                  GOOGA215   858
      GO TO 30                                                       GOOGA216   859
   31 ICARD=ICARD+1                                                  GOOGA217   860
C LEFT-ADJUST FIRST FIELD.                                           GOOGA218   861
      DO 1600 I=1,8                                                  GOOGA219   862
      IF(A(I).NE.BLANK)  GO TO 1610                                  GOOGA220   863
 1600 CONTINUE                                                       GOOGA221   864
      GO TO 30                                                       GOOGA222   865
 1610 IF(I.EQ.1) GO TO 1650                                          GOOGA223   866
      J=I-1                                                          GOOGA224   867
      K=8-J                                                          GOOGA225   868
      DO 1620 I=1,K                                                  GOOGA226   869
      A(I)=A(I+J)                                                    GOOGA227   870
 1620 A(I+J)=BLANK                                                   GOOGA228   871
 1650 CONTINUE                                                       GOOGA229   872
C LOOK FOR SEQSP CARD.                                               GOOGA230   873
      IF(A(1).EQ.S.AND.A(2).EQ.E.AND.A(3).EQ.Q.AND.A(4).EQ.G)IPARAM(7)=4GOOGA231   874
C LOOK FOR COMMENT CARD.                                             GOOGA232   875
      IF(A(1).EQ.DOLLAR.AND.KA.EQ.1)GO TO 35                         GOOGA233   876
C LOOK FOR ENDDATA CARD.                                             GOOGA234   877
      I=0                                                            GOOGA235   878
   32 I=I+1                                                          GOOGA236   879
      IF(I.GT.75)GO TO 35                                            GOOGA237   880
      IF(A(I).EQ.BLANK)GO TO 32                                      GOOGA238   881
      IF(A(I).NE.E)GO TO 40                                          GOOGA239   882
      IF(A(I+1).NE.N)GO TO 40                                        GOOGA240   883
      IF(A(I+2).NE.D)GO TO 40                                        GOOGA241   884
      IF(A(I+3).NE.D)GO TO 40                                        GOOGA242   885
C LEFT-ADJUST ENDDATA CARD.                                          GOOGA243   886
      K=73-I                                                         GOOGA244   887
      DO 33 J=1,72                                                   GOOGA245   888
      IF(J.LE.K)A(J)=A(J+I-1)                                        GOOGA246   889
   33 IF(J.GT.K)A(J)=BLANK                                           GOOGA247   890
      WRITE(NOUT,10)A                                                GOOGA248   891
      RETURN                                                         GOOGA249   892
   35 WRITE(NOUT,10)A                                                GOOGA250   893
      GO TO 30                                                       GOOGA251   894
C DETERMINE IF CARD IS TO BE PROCESSED.                              GOOGA252   895
   40 IF(KA.EQ.1)GO TO 150                                           GOOGA253   896
      IF(A(1).EQ.C.OR.A(1).EQ.G)GO TO 150                            GOOGA254   897
      IF(A(1).EQ.M.AND.A(2).EQ.P)GO TO 150                           GOOGA255   898
      NCOL=8                                                         GOOGA256   899
      IF(A(1).EQ.ASTER)GO TO 50                                      GOOGA257   900
```

```
      IF(A(1).EQ.PLUS)GO TO 60                                          GOOGA258   901
      GO TO 30                                                          GOOGA259   902
   50 NCOL=16                                                           GOOGA260   903
   60 NFIELD=64/NCOL                                                    GOOGA261   904
      I=0                                                               GOOGA262   905
   70 I=I+1                                                             GOOGA263   906
      IF(I.GT.NFIELD)GO TO 150                                          GOOGA264   907
      IPROC=0                                                           GOOGA265   908
      IFLAG=0                                                           GOOGA266   909
      J=0                                                               GOOGA267   910
   80 J=J+1                                                             GOOGA268   911
      IF(IPROC.EQ.1)GO TO 70                                            GOOGA269   912
      IF(J.LE.NCOL)GO TO 90                                             GOOGA270   913
      IF(IFLAG.EQ.1)GO TO 30                                            GOOGA271   914
      GO TO 70                                                          GOOGA272   915
   90 ICOL=8*NCOL*(I-1)+J                                               GOOGA273   916
      IF(A(ICOL).EQ.BLANK)GO TO 80                                      GOOGA274   917
      IFLAG=1                                                           GOOGA275   918
      DO 100 L=1,10                                                     GOOGA276   919
  100 IF(A(ICOL).EQ.ANUM(L))IPROC=1                                     GOOGA277   920
      GO TO 80                                                          GOOGA278   921
C PROCESS FIRST FIELD.                                                  GOOGA279   922
  150 NCOL=8                                                            GOOGA280   923
      KAST=8                                                            GOOGA281   924
      KBLK=8                                                            GOOGA282   925
      DO 160 I=1,8                                                      GOOGA283   926
      IF(A(I).NE.BLANK.AND.A(I).NE.ASTER.AND.A(I+1).EQ.BLANK)KBLK=I+1   GOOGA284   927
      IF(A(I).EQ.ASTER)KAST=I                                           GOOGA285   928
  160 IF(A(I).EQ.ASTER)NCOL=16                                          GOOGA286   929
      IF(A(1).EQ.PLUS)NCOL=8                                            GOOGA287   930
      IF(NCOL.EQ.16)GO TO 170                                           GOOGA288   931
      IF(KB.EQ.2)GO TO 200                                              GOOGA289   932
      IF(A(1).NE.PLUS)A(KBLK)=ASTER                                     GOOGA290   933
      IF(A(1).EQ.PLUS)A(1)=ASTER                                        GOOGA291   934
      GO TO 200                                                         GOOGA292   935
  170 IF(A(1).EQ.ASTER)GO TO 200                                        GOOGA293   936
      IA=MIN0(KAST,KBLK)                                                GOOGA294   937
      IB=MAX0(KAST,KBLK)                                                GOOGA295   938
      A(IB)=BLANK                                                       GOOGA296   939
      A(IA)=ASTER                                                       GOOGA297   940
C RIGHT-ADJUST ALL BULK DATA WHICH IS TO BE PROCESSED.                  GOOGA298   941
  200 NFIELD=64/NCOL                                                    GOOGA299   942
      IFIELD=0                                                          GOOGA300   943
  210 IFIELD=IFIELD+1                                                   GOOGA301   944
      IF(IFIELD.GT.NFIELD)GO TO 300                                     GOOGA302   945
      I=0                                                               GOOGA303   946
  220 I=I+1                                                             GOOGA304   947
      IF(I.GT.NCOL)GO TO 210                                            GOOGA305   948
      ICOL=9*NCOL*IFIELD-I                                              GOOGA306   949
      IF(A(ICOL).EQ.BLANK)GO TO 220                                     GOOGA307   950
      NBLANK=I-1                                                        GOOGA308   951
      NN=NCOL-NBLANK                                                    GOOGA309   952
      DO 230 I=1,NCOL                                                   GOOGA310   953
      J=9*NCOL*IFIELD-I                                                 GOOGA311   954
      JNB=J-NBLANK                                                      GOOGA312   955
      IF(I.LE.NN)A(J)=A(JNB)                                            GOOGA313   956
      IF(I.GT.NN)A(J)=BLANK                                             GOOGA314   957
  230 CONTINUE                                                          GOOGA315   958
      GO TO 210                                                         GOOGA316   959
C WRITE NEW CARD.                                                       GOOGA317   960
  300 IF(KB.EQ.1) A(73)=ASTER                                           GOOGA318   961
      IF(NCOL.EQ.8.AND.KB.EQ.1)GO TO 310                                GOOGA319   962
      WRITE(NOUT,10)A                                                   GOOGA320   963
      GO TO 30                                                          GOOGA321   964
  310 WRITE(NOUT,11)(A(I),I=1,40),ICARD,ICARD,(A(I),I=41,80)            GOOGA322   965
      GO TO 30                                                          GOOGA323   966
      END                                                               GOOGA324   967
      SUBROUTINE GRID(NGRID)                                            GRID    2  968
C PARTITION EXPANDABLE CORE.                                            GRID    3  969
      COMMON /BITS/ NBITIN,NBITEX                                       GRID    4  970
      COMMON /A/ MAXGRD,MAXDEG                                          GRID    5  971
      COMMON /X/ II(7),KOR                                              GRID    6  972
      MAX=16384                                                         GRID    7  973
      N=NGRID                                                           GRID    8  974
      NBITIN = 60                                                       JJ     20  975
      IF(N.LT.100) N=100                                                GRID   11  976
      IF(N.GT.MAX) GO TO 40                                             GRID   12  977
C CALCULATE WIDTH II(2) OF IG MATRIX.                                   GRID   13  978
20    L=60/NBITIN                                                       GRID   14  979
      M=60/NBITEX                                                       GRID   15  980
      N=N+L*M-1                                                         GRID   16  981
      N=N-MOD(N,L*M)                                                    GRID   17  982
      MAXGRD=N                                                          GRID   18  983
C  I=PACKED LENGTH FOR INTERNAL NUMBER.                                 GRID   19  984
C  J=PACKED LENGTH FOR ORIGINAL NUMBER.                                 GRID   20  985
      I=N/L                                                             GRID   21  986
      J=N/M                                                             GRID   22  987
C SET UP DIMENSIONS IN II ARRAY, WHERE IG(II1,II2),INV(II3,2),          GRID   23  988
C   INT(II4),ICC(II5),ILC(II6),NORIG(II7)                               GRID   24  989
      II(1)=I                                                           GRID   25  990
      II(3)=2*J                                                         GRID   26  991
      II(4)=J                                                           GRID   27  992
      II(5)=J                                                           GRID   28  993
      II(6)=J                                                           GRID   29  994
      II(7)=J                                                           GRID   30  995
      I=2*II(3)+II(4)+II(5)+II(6)+II(7)                                 GRID   31  996
      II(2)=(KOR-I)/(II(1)+2)                                           GRID   32  997
C DENOMINATOR CONTAINS A 2 TO ALLOW FOR 2 SCRATCH ARRAYS, EACH OF       GRID   33  998
C          LENGTH MAXDEG.                                               GRID   34  999
      II(2)=MIN0(II(2),N-1)                                             GRID   35 1000
```

```
      MAXDEG=II(2)                                                      GRID  36        1001
      RETURN                                                            GRID  37        1002
C SUBSTITUTE MAX IF NGRID TOO LARGE.                                    GRID  38        1003
40    N=MAX                                                             GRID  39        1004
      WRITE(6,50) NGRID,N                                               GRID  40        1005
   50 FORMAT(23H1BANDIT WARNING MESSAGE/10X,6H$GRID ,I10,5X,            GRID  41        1006
     +       9HTOO LARGE /10X,6H$GRID ,I10,5X,12HSUBSTITUTED. )         GRID  42        1007
      GO TO 20                                                          GRID  43        1008
      END                                                               GRID  44        1009
      SUBROUTINE READIT(A,IP,NIP)                                       READIT 2        1010
C THIS ROUTINE READS AND STORES (IN IP) NUMERIC DATA APPEARING ON       READIT 3        1011
C    $-CONTROL CARDS UP TO COLUMN 72.                                   READIT 4        1012
      DIMENSION ANUM(10)                                                READIT 5        1013
      DIMENSION A(1),IP(1)                                              READIT 6        1014
      DATA ANUM/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/                READIT 7        1015
C INITIALIZE ARRAY.                                                     READIT 8        1016
      NIP=0                                                             READIT 9        1017
      DO 10 I=1,40                                                      READIT10        1018
   10 IP(I)=0                                                           READIT11        1019
      I=3                                                               READIT12        1020
      DO 70 KOUNT=1,40                                                  READIT13        1021
      NUM=0                                                             READIT14        1022
      NUMFL=0                                                           READIT15        1023
   20 I=I+1                                                             READIT16        1024
      IF(I.LE.72) GO TO 30                                              READIT17        1025
      IF(NUMFL.EQ.1) GO TO 60                                           READIT18        1026
      RETURN                                                            READIT19        1027
   30 K=99                                                              READIT20        1028
      DO 40 J=1,10                                                      READIT21        1029
   40 IF(A(I).EQ.ANUM(J)) K=J-1                                         READIT22        1030
      IF(K.NE.99) GO TO 50                                              READIT23        1031
      IF(NUMFL) 60,20,60                                                READIT24        1032
   50 NUMFL=1                                                           READIT25        1033
      NUM=10*NUM+K                                                      READIT26        1034
      GO TO 20                                                          READIT27        1035
   60 NIP=KOUNT                                                         READIT28        1036
      IP(NIP)=NUM                                                       READIT29        1037
   70 CONTINUE                                                          READIT30        1038
      NIP=40                                                            READIT31        1039
      RETURN                                                            READIT32        1040
      END                                                               READIT33        1041
      SUBROUTINE BOMBIT(IERR)                                           BOMBIT 2        1042
C BOMB BANDIT TO SUPPRESS THE EXECUTION OF NASTRAN.                     BOMBIT 3        1043
      COMMON /B/ IPARAM(20)                                             BOMBIT 4        1044
      COMMON /K/ II(7),KORE,IFL                                         BOMBIT 5        1045
   3  FORMAT(43H1INSUFFICIENT CORE OR $GRID N CARD REQUIRED)            JJ    21        1046
    5 FORMAT(200(1H+,130X/))                                            BOMBIT15        1047
      CALL REMARK(40H   **************************************** )      BOMBIT16        1048
      GO TO (10,20,30,40,50,60,70,80,90), IERR                          BOMBIT17        1049
C EOF ENCOUNTERED.                                                      BOMBIT18        1050
   10 WRITE(6,12)                                                       BOMBIT19        1051
   12 FORMAT(55H1BANDIT FATAL ERROR -     MISSING BEGIN BULK OR ENDDATA,BOMBIT20        1052
     +    6H CARD. )                                                    BOMBIT21        1053
      CALL REMARK(39H   **MISSING BEGIN BULK OR ENDDATA CARD )          BOMBIT22        1054
      GO TO 500                                                         BOMBIT23        1055
C BULK DATA CARD OUT OF SORT.                                           BOMBIT24        1056
   20 CALL REMARK(31H   **BULK DATA CARD OUT OF SORT )                  BOMBIT25        1057
      GO TO 500                                                         BOMBIT26        1058
C SEQGP CARDS IN DECK AND RESEQUENCING REQUESTED.                       BOMBIT27        1059
   30 CALL REMARK(32H   **SEQGP CARDS ALREADY IN DECK )                 BOMBIT28        1060
      GO TO 500                                                         BOMBIT29        1061
C $SCHEME ILLEGAL ARGUMENTS.                                            BOMBIT30        1062
   40 WRITE(6,42)                                                       BOMBIT31        1063
   42 FORMAT(46H1BANDIT FATAL ERROR -     ILLEGAL ARGUMENTS ON,         BOMBIT32        1064
     +   14H $SCHEME CARD. )                                            BOMBIT33        1065
      CALL REMARK(30H   **ILLEGAL $SCHEME ARGUMENTS )                   BOMBIT34        1066
      GO TO 500                                                         BOMBIT35        1067
C  TOO MANY TERMS IN MPC EQUATION.                                      BOMBIT36        1068
   50 CALL REMARK(36H   **MPC EQUATION HAS TOO MANY TERMS )             BOMBIT37        1069
      GO TO 500                                                         BOMBIT38        1070
C MAXDEG EXCEEDED.                                                      BOMBIT39        1071
   60 CALL REMARK(28H   **MAXIMUM DEGREE EXCEEDED )                     BOMBIT40        1072
      WRITE(6,3)                                                        JJ    22        1073
      GO TO 500                                                         BOMBIT42        1074
C MAXGRD EXCEEDED.                                                      BOMBIT43        1075
   70 CALL REMARK(39H   **MAX NUMBER OF GRID POINTS EXCEEDED )          BOMBIT44        1076
      WRITE(6,3)                                                        JJ    23        1077
      GO TO 500                                                         BOMBIT46        1078
C NON-EXISTENT GRID POINT REFERENCE ON $-CARD                           BOMBIT47        1079
   80 CALL REMARK(32H   **ILLEGAL REFERENCE ON $-CARD )                 BOMBIT48        1080
      GO TO 500                                                         BOMBIT49        1081
C TOO MANY GRID POINTS ON $-CARD.                                       BOMBIT50        1082
   90 WRITE(6,92)                                                       BOMBIT51        1083
   92 FORMAT(51H1BANDIT FATAL ERROR -     TOO MANY POINTS ON $-CARD)    BOMBIT52        1084
      CALL REMARK(30H   **TOO MANY POINTS ON $-CARD )                   BOMBIT53        1085
      GO TO 500                                                         BOMBIT54        1086
C ABORT BANDIT.                                                         BOMBIT55        1087
  500 CALL REMARK(17H   **BANDIT ABORT )                                BOMBIT56        1088
      CALL REMARK(23H   **NASTRAN SUPPRESSED )                          BOMBIT57        1089
      CALL REMARK(40H   **************************************** )      BOMBIT58        1090
      WRITE(6,5)                                                        BOMBIT59        1091
      STOP                                                              BOMBIT66        1092
      END                                                               BOMBIT67        1093
      SUBROUTINE SCAT(KG,NCON,NEW,INV,II3,NORIG)                        SCAT   2        1094
C THIS ROUTINE USES SCATTER SORT TECHNIQUES FOR EACH GRID POINT         SCAT   3        1095
C          ENCOUNTERED TO DETERMINE WHETHER OR NOT THE POINT HAS        SCAT   4        1096
C          BEEN SEEN BEFORE.  IF NOT, INV, NORIG, AND NEW ARE UPDATED.  SCAT   5        1097
C INV(I,1) CONTAINS AN ORIGINAL GRID POINT NUMBER                       SCAT   6        1098
C INV(I,2) CONTAINS THE INTERNAL NUMBER ASSIGNED TO IT (BEFORE SORTING) SCAT   7        1099
      DIMENSION INV(II3,2),NORIG(1)                                     SCAT   8        1100
```

```
      COMMON /A/ MAXGRD,MAXDEG,KMOD                                    SCAT   9      1101
      DIMENSION KG(1)                                                  SCAT  10      1102
      DO 100 I=1,NCON                                                  SCAT  11      1103
      NOLD=KG(I)                                                       SCAT  12      1104
      IF(NOLD.EQ.0)GO TO 100                                           SCAT  13      1105
      LOC=NOLD-1                                                       SCAT  14      1106
   10 LOC=MOD(LOC,KMOD)+1                                              SCAT  15      1107
20    IF(INV(LOC,1).NE.0) GO TO 30                                     SCAT  16      1108
      INV(LOC,1)=NOLD                                                  SCAT  17      1109
      NEW=NEW+1                                                        SCAT  18      1110
      IF(NEW.GT.MAXGRD) GO TO 150                                      SCAT  19      1111
      NORIG(NEW)=NOLD                                                  SCAT  20      1112
      INV(LOC,2)=NEW                                                   SCAT  21      1113
      GO TO 40                                                         SCAT  22      1114
30    IF(INV(LOC,1).NE.NOLD) GO TO 10                                  SCAT  23      1115
40    KG(I)=INV(LOC,2)                                                 SCAT  24      1116
  100 CONTINUE                                                         SCAT  25      1117
      RETURN                                                           SCAT  26      1118
  150 WRITE(6,160) MAXGRD                                              SCAT  27      1119
  160 FORMAT(35H1 THIS STRUCTURE CONTAINS MORE THAN,I6,                SCAT  28      1120
     + 14H  GRID POINTS. /14H  FATAL ERROR. )                          SCAT  29      1121
      CALL DUMBIT(7)                                                   SCAT  30      1122
      END                                                              SCAT  31      1123
      SUBROUTINE BRIGIT(IG,II1,INV,II3,INT,ICC,NORIG,IP)               BRIGIT 2      1124
C THIS ROUTINE GENERATES A NEW INTERNAL/EXTERNAL CORRESPONDENCE        BRIGIT 3      1125
C      TABLE NORIG AND CONNECTION TABLE IG SUCH THAT THE NEW INTERNAL  BRIGIT 4      1126
C      NUMBERS CORRESPOND TO A SORT OF THE ORIGINAL NUMBERS INTO       BRIGIT 5      1127
C      ASCENDING ORDER.                                                BRIGIT 6      1128
C INPUT - IG,INV,NORIG                                                 BRIGIT 7      1129
C OUTPUT - IG,NORIG,ICC                                                BRIGIT 8      1130
C SCRATCH - INT,IP                                                     BRIGIT 9      1131
      DIMENSION IG(II1,1),INV(II3,2)                                   BRIGIT10      1132
      DIMENSION INT(1),ICC(1),NORIG(1),IP(1)                           BRIGIT11      1133
      COMMON /S/ NN,MM,IH,I3                                           BRIGIT12      1134
      COMMON /A/ MAXGRD,MAXDEG,KMOD,NMPC                               BRIGIT13      1135
      COMMON /BITS/ NBITIN,NBITEX,IPASS                                BRIGIT14      1136
      REWIND 8                                                         BRIGIT15      1137
C PERFORM A ROUGH SORT OF THE ORIGINAL GRID NUMBERS.                   BRIGIT16      1138
      L=0                                                              BRIGIT17      1139
      KFAC=-1                                                          BRIGIT18      1140
   20 KFAC=KFAC+1                                                      BRIGIT19      1141
      MIN=2147483647                                                   BRIGIT20      1142
      DO 50 I=1,KMOD                                                   BRIGIT21      1143
      IF(INV(I,1).GT.(KFAC*KMOD))                                      BRIGIT22      1144
     +       MIN=MIN0(MIN,INV(I,1))                                    BRIGIT23      1145
   50 CONTINUE                                                         BRIGIT24      1146
      KFAC=(MIN-1)/KMOD                                                BRIGIT25      1147
      DO 80 I=1,KMOD                                                   BRIGIT26      1148
      IS=INV(I,1)                                                      BRIGIT27      1149
      IF(IS.LE.(KFAC*KMOD).OR.IS.GT.(KFAC+1)*KMOD)GO TO 80             BRIGIT28      1150
      L=L+1                                                            BRIGIT29      1151
      INT(L)=INV(I,1)                                                  BRIGIT30      1152
   80 CONTINUE                                                         BRIGIT31      1153
      IF(L.LT.NN)GO TO 20                                              BRIGIT32      1154
C COMPLETE THE SORTING OF THE ORIGINAL GRID NUMBERS.                   BRIGIT33      1155
      CALL SORT(INT,NN)                                                BRIGIT34      1156
C DETERMINE CORRESPONDENCE (ICC) BETWEEN NORIG AND INT ARRAYS.         BRIGIT35      1157
      DO 130 I=1,NN                                                    BRIGIT36      1158
      L=INT(I)                                                         BRIGIT37      1159
      LOC=L-1                                                          BRIGIT38      1160
  110 LOC=MOD(LOC,KMOD)+1                                              BRIGIT39      1161
120   IF(INV(LOC,1).NE.L) GO TO 110                                    BRIGIT40      1162
      M=INV(LOC,2)                                                     BRIGIT41      1163
      ICC(M)=I                                                         BRIGIT42      1164
  130 CONTINUE                                                         BRIGIT43      1165
C TRANSFER INT ARRAY TO NORIG ARRAY.                                   BRIGIT44      1166
      DO 220 I=1,NN                                                    BRIGIT45      1167
  220 NORIG(I)=INT(I)                                                  BRIGIT46      1168
C CHANGE IG MATRIX ACCORDING TO CORRESPONDENCE TABLE ICC.              BRIGIT47      1169
      CALL SWITCH(IG,II1,INT,ICC,IP(1),IP(MAXDEG+1))                   BRIGIT48      1170
      REWIND 8                                                         BRIGIT49      1171
      RETURN                                                           BRIGIT50      1172
      END                                                              BRIGIT51      1173
      SUBROUTINE SORT(LIST,NL)                                         SORT   2      1174
C THIS SUBROUTINE SORTS A LIST OF LENGTH NL AND IS BIASED TOWARDS THOSE SORT   3      1175
C   LISTS NOT BADLY OUT OF SORT.                                       SORT   4      1176
      DIMENSION LIST(1)                                                SORT   5      1177
      IF(NL.LE.1) RETURN                                               SORT   6      1178
      NL1=NL-1                                                         SORT   7      1179
      DO 20 I=1,NL1                                                    SORT   8      1180
      K=NL-I                                                           SORT   9      1181
      KFLAG=0                                                          SORT  10      1182
      DO 10 J=1,K                                                      SORT  11      1183
      IF(LIST(J).LE.LIST(J+1)) GO TO 10                                SORT  12      1184
      KFLAG=1                                                          SORT  13      1185
      L=LIST(J)                                                        SORT  14      1186
      LIST(J)=LIST(J+1)                                                SORT  15      1187
      LIST(J+1)=L                                                      SORT  16      1188
   10 CONTINUE                                                         SORT  17      1189
      IF(KFLAG.EQ.0) RETURN                                            SORT  18      1190
   20 CONTINUE                                                         SORT  19      1191
      RETURN                                                           SORT  20      1192
      END                                                              SORT  21      1193
      SUBROUTINE SETIG(KG1,KG2,IG,II1,NORIG)                           SETIG  2      1194
C THIS ROUTINE SETS IG(KG1,-)=KG2 AND IG(KG2,-)=KG1 IF THIS            SETIG  3      1195
C          CONNECTION HAS NOT ALREADY BEEN SET.                        SETIG  4      1196
      DIMENSION IG(II1,1),NORIG(1)                                     SETIG  5      1197
      COMMON /S/ NN,MM,IH,I8                                           SETIG  6      1198
      COMMON /A/ MAXGRD,MAXDEG,KMOD,NMPC                               SETIG  7      1199
      COMMON /BITS/ NBITIN,NBITEX,IPASS                                SETIG  8      1200
```

```
      IF(KG1.EQ.0)RETURN                                                SETIG  9        1201
      IF(KG2.EQ.0)RETURN                                                SETIG 10        1202
      IF(KG1.EQ.KG2)RETURN                                              SETIG 11        1203
      DO 50 LOOP=1,2                                                    SETIG 12        1204
      L=KG1                                                             SETIG 13        1205
      K=KG2                                                             SETIG 14        1206
      IF(LOOP.EQ.1) GO TO 20                                            SETIG 15        1207
      L=KG2                                                             SETIG 16        1208
      K=KG1                                                             SETIG 17        1209
   20 M=0                                                               SETIG 18        1210
   30 M=M+1                                                             SETIG 19        1211
      IF(M.GT.MAXDEG) GO TO 60                                          SETIG 20        1212
      IS = IG(L,M)                                                      JJ    24        1213
      IF(IS.EQ.0) GO TO 40                                              SETIG 22        1214
      IF(IS.NE.K) GO TO 30                                              SETIG 23        1215
      GO TO 50                                                          SETIG 24        1216
   40 IG(L,M) = K                                                       JJ    25        1217
      MM=MAX0(MM,M)                                                     SETIG 26        1218
   50 CONTINUE                                                          SETIG 27        1219
      RETURN                                                            SETIG 28        1220
   60 WRITE(6,70) NORIG(L),MAXDEG                                       SETIG 29        1221
   70 FORMAT(12H1 GRID POINT,I12,26H   HAS DEGREE GREATER THAN,I6/      SETIG 30        1222
     + 14H  FATAL ERROR. )                                              SETIG 31        1223
      CALL BOMBIT(6)                                                    SETIG 32        1224
      END                                                               SETIG 33        1225
      SUBROUTINE TIGER(NEQ,IG,II1,LIST,NORIG)                           TIGER  2        1226
C THIS ROUTINE MAKES ADDITIONS TO THE CONNECTION TABLE IG TO REFLECT     TIGER  3        1227
C     THE PRESENCE OF MPC'S AND STORES THE DEPENDENT POINTS IN LIST.     TIGER  4        1228
C NEQ=NUMBER OF MPC EQUATIONS.                                          TIGER  5        1229
      DIMENSION IG(II1,1),LIST(1),NORIG(1)                              TIGER  6        1230
      COMMON /S/ NN,MM,IH,IB                                            TIGER  7        1231
      COMMON /A/ MAXGRD,MAXDEG,KMOD,NMPC                                TIGER  8        1232
      COMMON /BITS/ NBITIN,NBITEX,IPASS                                 TIGER  9        1233
      DIMENSION KG(40)                                                  TIGER 10        1234
      IF(NEQ.EQ.0)RETURN                                                TIGER 11        1235
      REWIND 11                                                         TIGER 12        1236
C INITIALIZE LIST.                                                      TIGER 13        1237
      DO 20 I=1,NN                                                      TIGER 14        1238
   20 LIST(I)=0                                                         TIGER 15        1239
C GENERATE NEW CONNECTIONS.                                             TIGER 16        1240
      DO 100 II=1,NEQ                                                   TIGER 17        1241
      READ(11)KG                                                        TIGER 18        1242
      IGRID=KG(1)                                                       TIGER 19        1243
      LIST(IGRID)=IGRID                                                 TIGER 20        1244
      DO 100 I=1,MAXDEG                                                 TIGER 21        1245
      L = IG(IGRID,I)                                                   JJ    26        1246
      DO 100 J=2,NMPC                                                   TIGER 23        1247
  100 CALL SETIG(L,KG(J),IG,II1,NORIG)                                  TIGER 24        1248
      REWIND 11                                                         TIGER 25        1249
      RETURN                                                            TIGER 26        1250
      END                                                               TIGER 27        1251
      SUBROUTINE SWITCH(IG,III,IFLAG,KT,KA,KB)                          SWITCH 2        1252
C THIS SUBROUTINE GENERATES A NEW IG MATRIX ACCORDING TO THE            SWITCH 3        1253
C      CORRESPONDENCE TABLE KT, WHICH MUST BE SET UP                    SWITCH 4        1254
C      PRIOR TO THE CALL.    ONLY INTERNAL NUMBERS ARE ALLOWED          SWITCH 5        1255
C      AS VALUES OF KT.                                                 SWITCH 6        1256
C                                                                       SWITCH 7        1257
C INPUT - IG,KT                                                         SWITCH 8        1258
C OUTPUT - IG                                                           SWITCH 9        1259
C SCRATCH - IFLAG,KA,KB                                                 SWITCH10        1260
C                                                                       SWITCH11        1261
      DIMENSION IG(III,1),IFLAG(1),KT(1),KA(1),KB(1)                    SWITCH12        1262
      COMMON /S/ NN,MM,IH,IB                                            SWITCH13        1263
      COMMON /A/ MAXGRD,MAXDEG,KMOD,NMPC                                SWITCH14        1264
      COMMON /BITS/ NBITIN,NBITEX,IPASS                                 SWITCH15        1265
C KT=CORRESPONDENCE TABLE.  KT(OLD) = NEW.                              SWITCH16        1266
C KA,KB = TEMPORARY STORAGE ROWS.                                       SWITCH17        1267
      DO 100 I=1,NN                                                     SWITCH18        1268
      DO 90 J=1,MM                                                      SWITCH19        1269
      L = IG(I,J)                                                       JJ    27        1270
      IF(L.LE.0) GO TO 100                                              SWITCH21        1271
      IS=KT(L)                                                          SWITCH22        1272
      IG(I,J) = IS                                                      JJ    28        1273
   90 CONTINUE                                                          SWITCH24        1274
  100 CONTINUE                                                          SWITCH25        1275
C INITIALIZE FLAGS.                                                     SWITCH26        1276
      DO 120 I=1,NN                                                     SWITCH27        1277
  120 IFLAG(I)=0                                                        SWITCH28        1278
C INITIALIZE TEMPORARY STORAGE ROWS.                                    SWITCH29        1279
      DO 130 I=1,MM                                                     SWITCH30        1280
      KA(I)=0                                                           SWITCH31        1281
  130 KB(I)=0                                                           SWITCH32        1282
C RE-ORDER ROWS OF IG MATRIX.                                           SWITCH33        1283
      DO 200 IROW=1,NN                                                  SWITCH34        1284
      IF(IFLAG(IROW).EQ.1) GO TO 200                                    SWITCH35        1285
      IF(KT(IROW).EQ.IROW) GO TO 200                                    SWITCH36        1286
      IFLAG(IROW)=1                                                     SWITCH37        1287
      DO 140 J=1,MM                                                     SWITCH38        1288
  140 KB(J) = IG(IROW,J)                                                JJ    29        1289
      L=KT(IROW)                                                        SWITCH40        1290
  150 IFLAG(L)=1                                                        SWITCH41        1291
      DO 160 J=1,MM                                                     SWITCH42        1292
      KA(J) = IG(L,J)                                                   JJ    30        1293
      IG(L,J) = KB(J)                                                   JJ    31        1294
  160 KB(J)=KA(J)                                                       SWITCH45        1295
      M=KT(L)                                                           SWITCH46        1296
      IF(IFLAG(M).EQ.1) GO TO 170                                       SWITCH47        1297
      L=M                                                               SWITCH48        1298
      GO TO 150                                                         SWITCH49        1299
  170 DO 180 J=1,MM                                                     SWITCH50        1300
```

```
  180 IG(M,J) = KB(J)                                                    JJ    32        1301
  200 CONTINUE                                                           SWITCH52        1302
      RETURN                                                             SWITCH53        1303
      END                                                                SWITCH54        1304
      SUBROUTINE MORRIS(LIST,NL,IG,II1)                                  MORRIS 2        1305
C THIS ROUTINE DELETES ALL REFERENCE IN THE CONNECTION TABLE IG          MORRIS 3        1306
C      TO THOSE POINTS IN A LIST OF LENGTH NL.                           MORRIS 4        1307
      DIMENSION IG(II1,1),LIST(1)                                        MORRIS 5        1308
      COMMON /S/ NN,MM                                                   MORRIS 6        1309
      COMMON /A/ MAXGRD                                                  MORRIS 7        1310
      COMMON /BITS/ NBITIN,NBITEX                                        MORRIS 8        1311
C COMPRESS OUT DUPLICATE ENTRIES IN LIST.                                MORRIS 9        1312
      CALL FIXIT(LIST,NL)                                                MORRIS10        1313
      IF(NL.LE.0) RETURN                                                 MORRIS11        1314
      MM1=MM-1                                                           MORRIS12        1315
      DO 60 IJ=1,NL                                                      MORRIS13        1316
      I=LIST(IJ)                                                         MORRIS14        1317
      DO 50 J=1,MM                                                       MORRIS15        1318
      L=IG(I,J)                                                          JJ    33        1319
      IF(L.EQ.0) GO TO 60                                                MORRIS17        1320
      K=0                                                                MORRIS18        1321
   20 K=K+1                                                              MORRIS19        1322
      M=IG(L,K)                                                          JJ    34        1323
      IF(M.NE.I) GO TO 20                                                MORRIS21        1324
      IF(K.GE.MM) GO TO 40                                               MORRIS22        1325
      DO 30 N=K,MM1                                                      MORRIS23        1326
      IS=IG(L,N+1)                                                       JJ    35        1327
   30 IG(L,N) = IS                                                       JJ    36        1328
   40 IG(L,MM) = 0                                                       JJ    37        1329
      IG (I,J) = 0                                                       JJ    38        1330
   50 CONTINUE                                                           MORRIS28        1331
   60 CONTINUE                                                           MORRIS29        1332
      RETURN                                                             MORRIS30        1333
      END                                                                MORRIS31        1334
      SUBROUTINE FIXIT(LIST,NL)                                          FIXIT  2        1335
C THIS ROUTINE COMPRESSES OUT ZEROES AND MULTIPLE ENTRIES IN A LIST      FIXIT  3        1336
C   ORIGINALLY OF LENGTH NL.    A CORRECTED LENGTH NL IS RETURNED TO     FIXIT  4        1337
C   THE CALLING PROGRAM.                                                 FIXIT  5        1338
      DIMENSION LIST(1)                                                  FIXIT  6        1339
      IF(NL.LE.0) RETURN                                                 FIXIT  7        1340
      IF(NL.EQ.1) GO TO 110                                              FIXIT  8        1341
      NL1=NL-1                                                           FIXIT  9        1342
C DELETE DUPLICATE ENTRIES.                                              FIXIT 10        1343
      DO 20 I=1,NL1                                                      FIXIT 11        1344
      IF(LIST(I).EQ.0) GO TO 20                                          FIXIT 12        1345
      I1=I+1                                                             FIXIT 13        1346
      DO 10 J=I1,NL                                                      FIXIT 14        1347
      IF(LIST(I).NE.LIST(J)) GO TO 10                                    FIXIT 15        1348
      LIST(I)=0                                                          FIXIT 16        1349
      GO TO 20                                                           FIXIT 17        1350
10    CONTINUE                                                           FIXIT 18        1351
20    CONTINUE                                                           FIXIT 19        1352
C DELETE ZEROES.                                                         FIXIT 20        1353
      DO 40 I=1,NL1                                                      FIXIT 21        1354
      K=0                                                                FIXIT 22        1355
25    IF(LIST(I).NE.0) GO TO 40                                          FIXIT 23        1356
      K=K+1                                                              FIXIT 24        1357
      DO 30 J=I,NL1                                                      FIXIT 25        1358
30    LIST(J)=LIST(J+1)                                                  FIXIT 26        1359
      LIST(NL)=0                                                         FIXIT 27        1360
      IF(K.GE.(NL-I+1)) GO TO 70                                         FIXIT 28        1361
      GO TO 25                                                           FIXIT 29        1362
40    CONTINUE                                                           FIXIT 30        1363
C CALCULATE NEW LENGTH NL.                                               FIXIT 31        1364
   70 DO 80 I=1,NL                                                       FIXIT 32        1365
      J=NL-I+1                                                           FIXIT 33        1366
      IF(LIST(J).NE.0) GO TO 90                                          FIXIT 34        1367
   80 CONTINUE                                                           FIXIT 35        1368
   90 NL=NL-I+1                                                          FIXIT 36        1369
      RETURN                                                             FIXIT 37        1370
  110 IF(LIST(1).EQ.0) NL=0                                              FIXIT 38        1371
      RETURN                                                             FIXIT 39        1372
      END                                                                FIXIT 40        1373
      SUBROUTINE SCHEME(NT,NUM,NCM,IO,IP,IG,II1,IC,IDEG,IDIS,IW,         SCHEME 2        1374
     + NEW,ICC,ILD,IPP)                                                  SCHEME 3        1375
C  IO IS VALID IFF 2.LE.IO.LE.3                                          SCHEME 4        1376
      DIMENSION IG(II1,1),IC(1),IDEG(1),IDIS(1),IW(1)                    SCHEME 5        1377
      DIMENSION NEW(1),ICC(1),ILD(1),IPP(1)                              SCHEME 6        1378
C IPP HAS DIMENSION 2*MAXDEG                                             SCHEME 7        1379
      COMMON /S/ NN,MM,IM,I8                                             SCHEME 8        1380
      COMMON /P/ IMO,IME                                                 SCHEME 9        1381
      COMMON /A/ MAXGRD                                                  SCHEME10        1382
      COMMON /C/ INAKN,LINE,KORIG,KNEW                                   SCHEME11        1383
      COMMON /BITS/ NBITIN,NBITEX,IPASS                                  SCHEME12        1384
      COMMON /TIME/ STIME,NCM                                            SCHEME13        1385
      COMMON /J/ IPARAM(20)                                              SCHEME14        1386
      COMMON /DDL/ ISTART(100),IGNORE(100),IFIRST(100)                   SCHEME15        1387
      COMMON /DDLL/ IDIM,ISTA,IIG,IFIR                                   SCHEME16        1388
      DIMENSION NODESL(100)                                              SCHEME17        1389
      EQUIVALENCE (IM,ATIME)                                             SCHEME18        1390
C     DETERMINE THE DEGREE OF EACH NODE.                                 SCHEME19        1391
      CALL DEGREE(IG,II1,IDEG)                                           SCHEME20        1392
C DETERMINE MODD, THE MOST PREVALENT NODAL DEGREE.                       SCHEME21        1393
      MODD=MODE(IDEG,IPP)                                                SCHEME22        1394
C     DETERMINE THE NUMBER OF COMPONENTS, NCM.                           SCHEME23        1395
      NCM=COMPNT(IG,II1,IC,IDEG,IW,ICC)                                  SCHEME24        1396
C     DETERMINE THE MAXIMUM DEGREE OF ANY NODE.                          SCHEME25        1397
      MAXD=MAXDGR(0,IC,IDEG)                                             SCHEME26        1398
      MM=MAXD                                                            SCHEME27        1399
C     DETERMINE THE ORIGINAL BANDWIDTH,IS.                               SCHEME28        1400
```

```
      DO 30 I=1,NN                                                    SCHEME29      1401
      NEW(I)=I                                                        SCHEME30      1402
   30 ILD(I)=I                                                        SCHEME31      1403
      IS=MAXBND(0,IG,II1,IC,IDEG,NEW,ILD)                            SCHEME32      1404
      KORIG=IS                                                        SCHEME33      1405
      IH0=IH                                                          SCHEME34      1406
C     INITIALIZE  NEW  AND  ILD  ARRAYS.                              SCHEME35      1407
      DO 35 I=1,NN                                                    SCHEME36      1408
      NEW(I)=0                                                        SCHEME37      1409
   35 ILD(I)=0                                                        SCHEME38      1410
C     IF IP IS NOT EQUAL TO 0, THEN PRINT COMPONENT NUMBER,DEGREE,    SCHEME39      1411
C     AND CONNECTIONS FOR EACH NODE.                                  SCHEME40      1412
      IF(IP.EQ.0) GO TO 31                                            SCHEME41      1413
C PRINT INTERNAL NUMBER CONNECTION TABLE.                             SCHEME42      1414
      DO 60 I=1,NN                                                    SCHEME43      1415
      IF(MOD(I,LINE).EQ.1)WRITE(6,19)                                 SCHEME44      1416
   19 FORMAT(37H1LABEL  COMP MDIST  DEGR  CONNECTIONS ,10X,           SCHEME45      1417
     1   18H(INTERNAL NUMBERS) )                                      SCHEME46      1418
      MDIST=0                                                         SCHEME47      1419
      DO 65 J=1,MAXD                                                  SCHEME48      1420
      IS1 = IG(I,J)                                                   JJ    39      1421
      IF(IS1.EQ.0)GO TO 65                                            SCHEME50      1422
      MDIST=MAXO(MDIST,IABS(I-IS1))                                   SCHEME51      1423
   65 CONTINUE                                                        SCHEME52      1424
      IPP(1)=IC(I)                                                    SCHEME53      1425
      IPP(2)=IDEG(I)                                                  SCHEME54      1426
      DO 610 IP1=1,MAXD                                               SCHEME55      1427
  610 IPP (IP1+2) = IG(I,IP1)                                         JJ    40      1428
      IS1=MAXD+2                                                      SCHEME57      1429
   60 WRITE(6,61) I,IPP(1),MDIST,(IPP(J),J=2,IS1)                     SCHEME58      1430
   61 FORMAT(5I6,20I5/ 25(25X,21I5/))                                 SCHEME59      1431
      WRITE(6,700)                                                    SCHEME60      1432
  700 FORMAT(1H1,//,32X,31HPROGRAMMER INFORMATION MESSAGES /)          SCHEME61      1433
      WRITE(6,29) IS,IH                                               SCHEME62      1434
   29 FORMAT(19H ORIGINAL BANDWIDTH,I7,10H   PROFILE,I10)             SCHEME63      1435
      WRITE(6,27) MODD                                                SCHEME64      1436
   27 FORMAT(30H MODE OF DEGREE DISTRIBUTION =,I5)                    SCHEME65      1437
      IF(ISTA.LE.0) GO TO 31                                          SCHEME66      1438
      WRITE(6,701)                                                    SCHEME67      1439
  701 FORMAT(34H STARTING NODES SUPPLIED BY USER -)                   SCHEME68      1440
      WRITE(6,100) (ISTART(I),I=1,ISTA)                               SCHEME69      1441
   31 CONTINUE                                                        JJ    41      1442
      IF(IO.EQ.3) IS=IH                                               SCHEME72      1443
C     GENERATE NUMBERING SCHEME FOR EACH COMPONENT, NC.               SCHEME73      1444
      DO 500 NC=1,NCM                                                 SCHEME74      1445
C     DETERMINE THE RANGE OF DEGREES (MI  TO  MAD) OF NODES OF INTEREST.SCHEME75    1446
      MI=MINDEG(NC,IC,IDEG)                                           SCHEME76      1447
      MAD=MI                                                          SCHEME77      1448
      IF(NOM) 90,87,90                                                SCHEME78      1449
   90 MA=MAXDGR(NC,IC,IDEG)                                           SCHEME79      1450
      MAD=MI+((MA-MI)*NUM)/NOM                                        SCHEME80      1451
C MAKE SURE THAT MAD IS LESS THAN MODD.                               SCHEME81      1452
      MAD=MIN0(MAD,MODD-1)                                            SCHEME82      1453
      MAD=MAX0(MAD,MI)                                                SCHEME83      1454
C     DETERMINE BANDWIDTH OR SUM CRITERION FOR EACH NODE MEETING SPECI- SCHEME84    1455
C             FIED CONDITION.                                         SCHEME85      1456
   87 IF(IP.EQ.0) GO TO 91                                            SCHEME86      1457
      WRITE(6,162) NC                                                 SCHEME87      1458
  162 FORMAT(22H *******     COMPONENT,I5,12H      *******)           SCHEME88      1459
      IF(IO.EQ.2) WRITE(6,169)                                        SCHEME89      1460
  169 FORMAT(43H OPTION 2 SELECTED  (CRITERION - BANDWIDTH ,          SCHEME90      1461
     +  57HMINIMIZATION;   CONDITION - MINMAX NUMBER OF NODES/LEVEL) ) SCHEME91     1462
      IF(IO.EQ.3) WRITE(6,179)                                        SCHEME92      1463
  179 FORMAT(52H OPTION 3 SELECTED (CRITERION - MINIMIZATION OF SUM;,  SCHEME93      1464
     +  44H   CONDITION - MINMAX NUMBER OF NODES/LEVEL) )             SCHEME94      1465
   91 CALL DIAM(NC,MAD,NL,NODESL,MAXLEV,IG,II1,IC,IDEG,IDIS,IW,ICC)   SCHEME95      1466
      IF(IP.EQ.0) GO TO 67                                            SCHEME96      1467
      WRITE(6,39) NC,MAD                                              SCHEME97      1468
      WRITE(6,59) MAXLEV                                              SCHEME98      1469
      WRITE(6,100) (NODESL(J),J=1,NL)                                 SCHEME99      1470
   67 CONTINUE                                                        SCHEM100      1471
      IF(ISTA.LE.0) GO TO 760                                         SCHEM101      1472
      M=0                                                             SCHEM102      1473
      DO 750 I=1,ISTA                                                 SCHEM103      1474
      J=ISTART(I)                                                     SCHEM104      1475
      IF(IC(J).NE.NC) GO TO 750                                       SCHEM105      1476
      M=M+1                                                           SCHEM106      1477
      DO 755 K=1,99                                                   SCHEM107      1478
      L=101-K                                                         SCHEM108      1479
  755 NODESL(L)=NODESL(L-1)                                           SCHEM109      1480
      NODESL(1)=J                                                     SCHEM110      1481
  750 CONTINUE                                                        SCHEM111      1482
      NL=MIN0(NL+M,100)                                               SCHEM112      1483
      CALL FIXIT(NODESL,NL)                                           SCHEM113      1484
  760 CONTINUE                                                        SCHEM114      1485
      IF(IP.EQ.0) GO TO 63                                            SCHEM115      1486
      IF(ISTA.LE.0) GO TO 63                                          SCHEM116      1487
      WRITE(6,730)                                                    SCHEM117      1488
  730 FORMAT(48H MERGED LIST OF STARTING NODES SUPPLIED BY USER ,     SCHEM118      1489
     +  15HAND BY BANDIT -)                                           SCHEM119      1490
      WRITE(6,100) (NODESL(I),I=1,NL)                                 SCHEM120      1491
   39 FORMAT(10H COMPONENT,I5,19H     MAX DEGREE USED,I5)             SCHEM121      1492
   59 FORMAT(52H STARTING NODES FOR MINMAX NUMBER OF NODES PER LEVEL,I5)SCHEM122     1493
  100 FORMAT(4X,20I5)                                                 SCHEM123      1494
   63 CONTINUE                                                        SCHEM124      1495
      JMAX=MIN0(NT,NL)                                                SCHEM125      1496
      IM=900000000                                                    SCHEM126      1497
      IMM=IM                                                          SCHEM127      1498
      DO 400 J=1,JMAX                                                 SCHEM128      1499
      CALL RELABL(1,NODESL(J ),IG,II1,IC,IDEG,IDIS,IW,NEW,ICC,ILD)    SCHEM129      1500
```

```
      IB=MAXBND(NC,IG,II1,IC,IDEG,NEW,ILD)                              SCHEM130   1501
      IF(IP.NE.0) WRITE(6,69) NODESL(J),IB,IH                           SCHEM131   1502
   69 FORMAT(14H STARTING NODE,I6,4X,9HBANDWIDTH,I6,3X,7HPROFILE,I8)    SCHEM132   1503
      IF(IO.EQ.3) IB=IH                                                 SCHEM133   1504
      IE=ICC(NC+1)-1                                                    SCHEM134   1505
      IF(IH-IB) 400,350,300                                             SCHEM135   1506
  300 IH=IB                                                             SCHEM136   1507
      IHH=IH                                                            SCHEM137   1508
      IJ=J                                                              SCHEM138   1509
      GO TO 400                                                         SCHEM139   1510
  350 IF(IHH.LE.IH) GO TO 400                                           SCHEM140   1511
      IHH=IH                                                            SCHEM141   1512
      IJ=J                                                              SCHEM142   1513
400   CONTINUE                                                          SCHEM143   1514
      CALL RELABL(1,NODESL(IJ),IG,II1,IC,IDEG,IDIS,IH,NEW,ICC,ILD)      SCHEM144   1515
  500 CONTINUE                                                          SCHEM145   1516
      CALL STACK(IDEG,NEW,ILD,IH)                                       SCHEM146   1517
      IB=MAXBND(0,IG,II1,IC,IDEG,NEW,ILD)                               SCHEM147   1518
      IF(IP.EQ.0) GO TO 710                                             SCHEM148   1519
      WRITE(6,705)                                                      SCHEM149   1520
  705 FORMAT(21H0 ORIGINAL LABELING -)                                  SCHEM150   1521
      WRITE(6,708) KORIG,IH0                                            SCHEM151   1522
      WRITE(6,707)                                                      SCHEM152   1523
  707 FORMAT(21H  STD CM RELABELING -)                                  SCHEM153   1524
      WRITE(6,708) IB,IH                                                SCHEM154   1525
  708 FORMAT(1H#,26X,9HBANDWIDTH,I7,10X,7HPROFILE,I10)                  SCHEM155   1526
  709 FORMAT(21H  REV CM RELABELING -)                                  SCHEM156   1527
  710 IF(IO.EQ.3) IB=IH                                                 SCHEM157   1528
C PROFILE = SUM CRIT                                                    SCHEM158   1529
C IS=ORIGINAL BANDWIDTH (OR SUM CRIT IF IO.EQ.3)                        SCHEM159   1530
C IB=CURRENT BANDWIDTH (OR SUM CRIT IF IO.EQ.3)                         SCHEM160   1531
C IH=CURRENT PROFILE, IH0=ORIGINAL PROFILE                              SCHEM161   1532
      IF(IB-IS) 715,742,744                                             SCHEM162   1533
  742 IF(IH.LT.IH0) GO TO 715                                           SCHEM163   1534
  744 DO 712 I=1,NN                                                     SCHEM164   1535
      ILD(I)=I                                                          SCHEM165   1536
  712 NEW(I)=I                                                          SCHEM166   1537
      CALL STACK(IDEG,NEW,ILD,IH)                                       SCHEM167   1538
      IB=IS                                                             SCHEM168   1539
      IH=IH0                                                            SCHEM169   1540
      IF(IP.EQ.0) GO TO 715                                             SCHEM170   1541
      WRITE(6,713)                                                      SCHEM171   1542
  713 FORMAT(21H ORIG CM RELABELING -)                                  SCHEM172   1543
      WRITE(6,708) IB,IH                                                SCHEM173   1544
  715 IHE=IH                                                            SCHEM174   1545
      CALL REVERS(NEW,ILD)                                              SCHEM175   1546
      IB=MAXBND(0,IG,II1,IC,IDEG,NEW,ILD)                               SCHEM176   1547
      IF(IP.EQ.0) GO TO 717                                             SCHEM177   1548
      WRITE(6,709)                                                      SCHEM178   1549
      WRITE(6,708) IB,IH                                                SCHEM179   1550
  717 IF(IH.LT.IHE) GO TO 720                                           SCHEM180   1551
      CALL REVERS(NEW,ILD)                                              SCHEM181   1552
      IB=MAXBND(0,IG,II1,IC,IDEG,NEW,ILD)                               SCHEM182   1553
  720 IHE=IH                                                            SCHEM183   1554
      KNEW=IB                                                           SCHEM184   1555
      IF(IP.EQ.0) GO TO 508                                             SCHEM185   1556
      WRITE(6,722)                                                      SCHEM186   1557
  722 FORMAT(21H  ** FINAL LABELING -)                                  SCHEM187   1558
      WRITE(6,708) KNEW,IHE                                             SCHEM188   1559
  503 CONTINUE                                                          JJ    42   1560
  600 RETURN                                                            SCHEM194   1561
      END                                                               SCHEM195   1562
      SUBROUTINE STACK(IDEG,NEW,ILD,IW)                                 STACK   2  1563
C STACK POINTS OF ZERO DEGREE AT END OF THE NUMBERING.                  STACK   3  1564
      DIMENSION IDEG(1),NEW(1),ILD(1),IW(1)                             STACK   4  1565
C IW IS SCRATCH STORAGE.                                                STACK   5  1566
      COMMON /S/ NN                                                     STACK   6  1567
      COMMON /ZERO/ KT                                                  STACK   7  1568
      KT=0                                                              STACK   8  1569
      NN1=NN-1                                                          STACK   9  1570
C LIST POINTS OF ZERO DEGREE AND INCREMENT COUNTER KT.                  STACK  10  1571
      DO 10 I=1,NN                                                      STACK  11  1572
      IF(IDEG(I).GT.0) GO TO 10                                         STACK  12  1573
      KT=KT+1                                                           STACK  13  1574
      IW(KT)=ILD(I)                                                     STACK  14  1575
   10 CONTINUE                                                          STACK  15  1576
      IF(KT.LE.0) GO TO 70                                              STACK  16  1577
C SORT LIST OF RENUMBERED NUMBERS TO BE STACKED.                        STACK  17  1578
      CALL SORT(IW,KT)                                                  STACK  18  1579
C STACK POINTS OF ZERO DEGREE AT END OF NEW.                            STACK  19  1580
      DO 40 L=1,KT                                                      STACK  20  1581
      I=IW(L)-L+1                                                       STACK  21  1582
      K=NEW(I)                                                          STACK  22  1583
      IF(I.GE.NN) GO TO 30                                              STACK  23  1584
      DO 20 J=I,NN1                                                     STACK  24  1585
   20 NEW(J)=NEW(J+1)                                                   STACK  25  1586
   30 NEW(NN)=K                                                         STACK  26  1587
   40 CONTINUE                                                          STACK  27  1588
C CORRECT ILD, THE INVERSE OF NEW.                                      STACK  28  1589
   70 DO 80 I=1,NN                                                      STACK  29  1590
      K=NEW(I)                                                          STACK  30  1591
   80 ILD(K)=I                                                          STACK  31  1592
      RETURN                                                            STACK  32  1593
      END                                                               STACK  33  1594
      SUBROUTINE REVERS(NEW,ILD)                                        REVERS  2  1595
C REVERSE THE NUMBERING OF THE FIRST NN-KT GRID POINTS.                 REVERS  3  1596
C NN=NUMBER OF GRID POINTS.                                             REVERS  4  1597
C KT=THE NUMBER OF POINTS OF ZERO DEGREE (STACKED AT END OF NEW         REVERS  5  1598
C       BY STACK)                                                       REVERS  6  1599
      DIMENSION NEW(1),ILD(1)                                           REVERS  7  1600
```

```
      COMMON /S/ NN                                                    REVERS 8      1601
      COMMON /ZERO/ KT                                                 REVERS 9      1602
C REVERSE NEW ARRAY.                                                   REVERS10      1603
      J=(NN-KT)/2                                                      REVERS11      1604
      LL=NN-KT+1                                                       REVERS12      1605
      DO 10 I=1,J                                                      REVERS13      1606
      L=LL-I                                                           REVERS14      1607
      K=NEW(L)                                                         REVERS15      1608
      NEW(L)=NEW(I)                                                    REVERS16      1609
   10 NEW(I)=K                                                         REVERS17      1610
C CORRECT ILD, THE INVERSE OF NEW.                                     REVERS18      1611
      DO 20 I=1,NN                                                     REVERS19      1612
      K=NEW(I)                                                         REVERS20      1613
   20 ILD(K)=I                                                         REVERS21      1614
      RETURN                                                           REVERS22      1615
      END                                                              REVERS23      1616
      SUBROUTINE DEGREE(IG,II1,IDEG)                                   DEGREE 2      1617
C SET UP THE IDEG ARRAY CONTAINING THE DEGREE OF EACH NODE STORED      DEGREE 3      1618
C     IN THE IG ARRAY.                                                 DEGREE 4      1619
C   IDEG(I)=DEGREE OF NODE I                                           DEGREE 5      1620
      DIMENSION IG(II1,1),IDEG(1)                                      DEGREE 6      1621
      COMMON /S/ NN,MM,IH,IB                                           DEGREE 7      1622
      COMMON /A/ MAXGRD                                                DEGREE 8      1623
      COMMON /BITS/ NBITIN,NBITEX,IPASS                                DEGREE 9      1624
      DO 100 I=1,NN                                                    DEGREE10      1625
      IDEG(I)=0                                                        DEGREE11      1626
      DO 80 J=1,MM                                                     DEGREE12      1627
      IF(IG(I,J)) 100,100,50                                           JJ    43      1628
   50 IDEG(I)=IDEG(I)+1                                                DEGREE14      1629
   80 CONTINUE                                                         DEGREE15      1630
  100 CONTINUE                                                         DEGREE16      1631
      RETURN                                                           DEGREE17      1632
      END                                                              DEGREE18      1633
      FUNCTION MODE(IDEG,MODD)                                         MODE   2      1634
C COMPUTE MODE, THE MOST PREVALENT NODAL DEGREE.    IF SEVERAL DEGREES MODE   3      1635
C      ARE EQUALLY PREVALENT, THE LOWEST IS CHOSEN.                    MODE   4      1636
      COMMON /S/ NN,MM                                                 MODE   5      1637
      DIMENSION IDEG(1),MODD(1)                                        MODE   6      1638
C IDEG(I)=DEGREE OF NODE I                                             MODE   7      1639
C  MODD(I)=NUMBER OF NODES OF DEGREE I                                 MODE   8      1640
      DO 10 I=1,MM                                                     MODE   9      1641
   10 MODD(I)=0                                                        MODE  10      1642
      DO 20 I=1,NN                                                     MODE  11      1643
      K=IDEG(I)                                                        MODE  12      1644
   20 MODD(K)=MODD(K)+1                                                MODE  13      1645
      MODE=0                                                           MODE  14      1646
      MAX=0                                                            MODE  15      1647
      DO 30 I=1,MM                                                     MODE  16      1648
      K=MODD(I)                                                        MODE  17      1649
      IF(K.LE.MAX) GO TO 30                                            MODE  18      1650
      MAX=K                                                            MODE  19      1651
      MODE=I                                                           MODE  20      1652
   30 CONTINUE                                                         MODE  21      1653
      RETURN                                                           MODE  22      1654
      END                                                              MODE  23      1655
      FUNCTION COMPNT(IG,II1,IC,IDEG,IW,ICC)                           COMPNT 2      1656
C THIS FUNCTION HAS AS ITS VALUE THE NUMBER OF COMPONENTS STORED       COMPNT 3      1657
C     IN THE CONNECTION ARRAY IG.                                      COMPNT 4      1658
C ALSO, IC AND ICC ARE SET UP.                                         COMPNT 5      1659
C   IC(I)=COMPONENT INDEX FOR NODE I                                   COMPNT 6      1660
C   ICC(I)=THE STARTING POSITION TO BE USED FOR LABELS IN COMPONENT I  COMPNT 7      1661
C THUS, ICC(I+1)-ICC(I)= THE NUMBER OF NODES IN COMPONENT I            COMPNT 8      1662
      DIMENSION IG(II1,1),IC(1),IDEG(1),IW(1),ICC(1)                   COMPNT 9      1663
      COMMON /S/ NN,MM,IH,IB                                           COMPNT10      1664
      COMMON /A/ MAXGRD                                                COMPNT11      1665
      COMMON /BITS/ NBITIN,NBITEX,IPASS                                COMPNT12      1666
C INITIALIZE ARRAYS.                                                   COMPNT13      1667
      DO 100 I=1,NN                                                    COMPNT14      1668
      ICC(I)=0                                                         COMPNT15      1669
      IC(I)=0                                                          COMPNT16      1670
  100 CONTINUE                                                         COMPNT17      1671
      NC=0                                                             COMPNT18      1672
      ICC(1)=1                                                         COMPNT19      1673
C CHECK IF IC IS COMPLETE.                                             COMPNT20      1674
  105 DO 110 I=1,NN                                                    COMPNT21      1675
      IF(IC(I)) 110,120,110                                            COMPNT22      1676
  110 COMPNT=NC                                                        COMPNT23      1677
      RETURN                                                           COMPNT24      1678
  120 NC=NC+1                                                          COMPNT25      1679
      KI=0                                                             COMPNT26      1680
      KO=1                                                             COMPNT27      1681
      IW(1)=I                                                          COMPNT28      1682
      IC(I)=NC                                                         COMPNT29      1683
      IF(NC-1)130,125,125                                              COMPNT30      1684
125   IS=ICC(NC)+1                                                     COMPNT31      1685
      ICC(NC+1)=IS                                                     COMPNT32      1686
  130 KI=KI+1                                                          COMPNT33      1687
      II=IW(KI)                                                        COMPNT34      1688
      N=IDEG(II)                                                       COMPNT35      1689
      IF(N)140,105,140                                                 COMPNT36      1690
  140 DO 200 I=1,N                                                     COMPNT37      1691
      IA = IG(II,I)                                                    JJ    44      1692
      IF(IC(IA)) 200,150,200                                           COMPNT39      1693
150   IC(IA)=NC                                                        COMPNT40      1694
      KO=KO+1                                                          COMPNT41      1695
      IW(KO)=IA                                                        COMPNT42      1696
      IS=ICC(NC+1)+1                                                   COMPNT43      1697
      ICC(NC+1)=IS                                                     COMPNT44      1698
  200 CONTINUE                                                         COMPNT45      1699
      IF(KO-KI)105,105,130                                             COMPNT46      1700
```

```
      END                                                               COMPNT47    1701
      FUNCTION MAXDGR(NC,IC,IDEG)                                       MAXDGR 2    1702
C THIS FUNCTION HAS AS ITS VALUE THE MAXIMUM DEGREE OF ANY NODE OF      MAXDGR 3    1703
C       COMPONENT NC IF NC.GT.0                                         MAXDGR 4    1704
C IF NC.LE.0, ALL COMPONENTS ARE CONSIDERED.                            MAXDGR 5    1705
      DIMENSION IC(1),IDEG(1)                                           MAXDGR 6    1706
      COMMON /S/ NN,MM,IH,IB                                            MAXDGR 7    1707
      M=0                                                               MAXDGR 8    1708
      DO 100 I=1,NN                                                     MAXDGR 9    1709
      IF(NC)40,50,40                                                    MAXDGR10    1710
40    IF(IC(I)-NC) 100,50,100                                           MAXDGR11    1711
50    IF(IDEG(I)-M) 100,100,60                                          MAXDGR12    1712
60    M=IDEG(I)                                                         MAXDGR13    1713
  100 CONTINUE                                                          MAXDGR14    1714
      MAXDGR=M                                                          MAXDGR15    1715
      RETURN                                                            MAXDGR16    1716
      END                                                               MAXDGR17    1717
      FUNCTION MAXBND(NC,IG,II1,IC,IDEG,NEW,ILD)                        MAXBND 2    1718
C THIS FUNCTION HAS AS ITS VALUE THE MAXIMUM DIFFERENCE BETWEEN NODE    MAXBND 3    1719
C     LABELS OF CONNECTED NODES FOR NODES OF COMPONENT NC.GT.0          MAXBND 4    1720
C IF NC.LE.0, ALL COMPONENTS ARE CONSIDERED.                            MAXBND 5    1721
C THE NODAL RENUMBERING DEFINED BY ILD AND NEW MUST BE SET UP PRIOR     MAXBND 6    1722
C       TO THE FUNCTION CALL.                                           MAXBND 7    1723
C COMPUTE IH, THE SUM CRIT (PROFILE).                                   MAXBND 8    1724
      DIMENSION IG(II1,1),IC(1),IDEG(1),NEW(1),ILD(1)                   MAXBND 9    1725
      COMMON /S/ NN,MM,IH,IB                                            MAXBND10    1726
      COMMON /A/ MAXGRD                                                 MAXBND11    1727
      COMMON /BITS/ NBITIN,NBITEX,IPASS                                 MAXBND12    1728
      IH=0                                                              MAXBND13    1729
      M=0                                                               MAXBND14    1730
      DO 100 I=1,NN                                                     MAXBND15    1731
      MX=0                                                              MAXBND16    1732
      IA=NEW(I)                                                         MAXBND17    1733
      IF(NC)40,50,40                                                    MAXBND18    1734
40    IF(IA.EQ.0)GO TO 100                                              MAXBND19    1735
      IF(NC-IC(IA)) 100,50,100                                          MAXBND20    1736
50    N=IDEG(IA)                                                        MAXBND21    1737
      IF(N)100,100,150                                                  MAXBND22    1738
150   DO 90 J=1,N                                                       MAXBND23    1739
      II = IG(IA,J)                                                     JJ    45    1740
      IB=MAX0(0,I-ILD(II))                                              MAXBND25    1741
      IF(IB.GT.MX) MX=IB                                                MAXBND26    1742
90    CONTINUE                                                          MAXBND27    1743
      IF(MX.GT.M) M=MX                                                  MAXBND28    1744
      IH=IH+MX                                                          MAXBND29    1745
100   CONTINUE                                                          MAXBND30    1746
      MAXBND=M                                                          MAXBND31    1747
      RETURN                                                            MAXBND32    1748
      END                                                               MAXBND33    1749
      FUNCTION MINDEG(NC,IC,IDEG)                                       MINDEG 2    1750
C THIS FUNCTION HAS AS ITS VALUE THE MINIMUM DEGREE OF ANY NODE OF      MINDEG 3    1751
C       COMPONENT NC IF NC.GT.0                                         MINDEG 4    1752
C IF NC.LE.0, ALL COMPONENTS ARE CONSIDERED.                            MINDEG 5    1753
      DIMENSION IC(1),IDEG(1)                                           MINDEG 6    1754
      COMMON /S/ NN,MM,IH,IB                                            MINDEG 7    1755
      M=10000                                                           MINDEG 8    1756
      DO 100 I=1,NN                                                     MINDEG 9    1757
      IF(NC)40,50,40                                                    MINDEG10    1758
40    IF(IC(I)-NC) 100,50,100                                           MINDEG11    1759
50    IF(M-IDEG(I)) 100,100,60                                          MINDEG12    1760
60    M=IDEG(I)                                                         MINDEG13    1761
  100 CONTINUE                                                          MINDEG14    1762
      MINDEG=M                                                          MINDEG15    1763
      RETURN                                                            MINDEG16    1764
      END                                                               MINDEG17    1765
      SUBROUTINE DIAM(NC,MAXDEG,NL,NODESL,MAXLEV,                       DIAM   2    1766
     + IG,II1,IC,IDEG,IDIS,IW,ICC)                                      DIAM   3    1767
C DETERMINE NL STARTING POINTS AND STORE IN NODESL.                     DIAM   4    1768
      DIMENSION IG(II1,1),IDIS(1),IW(1),ICC(1),IC(1),IDEG(1)            DIAM   5    1769
      COMMON /S/ NN,MM,IH,IB                                            DIAM   6    1770
      COMMON /A/ MAXGRD                                                 DIAM   7    1771
      COMMON /BITS/ NBITIN,NBITEX,IPASS                                 DIAM   8    1772
      DIMENSION NODESL(1)                                               DIAM   9    1773
      NL=0                                                              DIAM  10    1774
      MAXLEV=10000                                                      DIAM  11    1775
      DO 100 I=1,NN                                                     DIAM  12    1776
      IF(NC-IC(I)) 100,40,100                                           DIAM  13    1777
40    IF(MAXDEG-IDEG(I)) 100,105,105                                    DIAM  14    1778
  105 MD=IDIST(I,ML,MAXLEV,IG,II1,IC,IDEG,IDIS,IW,ICC)                  DIAM  15    1779
      IF(MD) 115,115,56                                                 DIAM  16    1780
   56 IF(ML-MAXLEV)58,64,100                                            DIAM  17    1781
   58 MAXLEV=ML                                                         DIAM  18    1782
      NL=1                                                              DIAM  19    1783
      NODESL(1)=I                                                       DIAM  20    1784
      GO TO 100                                                         DIAM  21    1785
   64 IF(NL.GE.100) GO TO 100                                           DIAM  22    1786
      NL=NL+1                                                           DIAM  23    1787
      NODESL(NL)=I                                                      DIAM  24    1788
  100 CONTINUE                                                          DIAM  25    1789
  110 RETURN                                                            DIAM  26    1790
  115 ML=1                                                              DIAM  27    1791
      NODESL(1)=I                                                       DIAM  28    1792
      MAXLEV=0                                                          DIAM  29    1793
      RETURN                                                            DIAM  30    1794
      END                                                               DIAM  31    1795
      SUBROUTINE RELABL(NS,NODES,IG,II1,IC,IDEG,IDIS,IW,NEW,ICC,ILD)    RELABL 2    1796
C GENERATE A RELABELING SCHEME STARTING WITH NS NODES FOR WHICH         RELABL 3    1797
C       LABELS HAVE BEEN STORED IN ARRAY NODES.                         RELABL 4    1798
C SET UP ILD AND NEW.                                                   RELABL 5    1799
C     ILD(OLD)=NEW                                                      RELABL 6    1800
```

```
C      NEW(NEW)=OLD,   THE INVERSE OF ILD                              RELABL 7        1801
       DIMENSION IG(II1,1),IC(1),IDEG(1),IDIS(1),IW(1),NEW(1),ICC(1)   RELABL 8        1802
       DIMENSION ILD(1)                                                RELABL 9        1803
       COMMON /S/ NN,MM,IH,IB                                          RELABL10        1804
       INTEGER X                                                       RELABL11        1805
       COMMON /A/ MAXGRD                                               RELABL12        1806
       COMMON /BITS/ NBITIN,NBITEX,IPASS                               RELABL13        1807
       DIMENSION NODES( 1),IAJ(50)                                     RELABL14        1808
       I=NODES(1)                                                      RELABL15        1809
       ICN=IC(I)                                                       RELABL16        1810
       NT=ICC(ICN)-1                                                   RELABL17        1811
       DO 50 I=1,NN                                                    RELABL18        1812
       IF(IC(I)-ICN) 50,40,50                                          RELABL19        1813
 40    IDIS(I)=0                                                       RELABL20        1814
 50    CONTINUE                                                        RELABL21        1815
       DO 100 J=1,NS                                                   RELABL22        1816
       JJ=NODES(J)                                                     RELABL23        1817
       IDIS(JJ)=-1                                                     RELABL24        1818
       JT=J+NT                                                         RELABL25        1819
       NEW(JT)=JJ                                                      RELABL26        1820
 100   ILD(JJ)=JT                                                     RELABL27        1821
       KI=NT                                                           RELABL28        1822
       KO=NS+NT                                                        RELABL29        1823
       LL=KO                                                           RELABL30        1824
       L=1                                                             RELABL31        1825
       J=KO                                                            RELABL32        1826
       NNC=ICC(ICN+1)-1                                                RELABL33        1827
 130   KI=KI+1                                                         RELABL34        1828
       IF(KI-LL)135,132,135                                            RELABL35        1829
 132   L=L+1                                                           RELABL36        1830
       LL=KO+1                                                         RELABL37        1831
 135   II=NEW(KI)                                                      RELABL38        1832
       N=IDEG(II)                                                      RELABL39        1833
       IF(N)140,255,140                                                RELABL40        1834
 140   IJ=0                                                            RELABL41        1835
       DO 200 I=1,N                                                    RELABL42        1836
       IA = IG(II,I)                                                   JJ    46        1837
       IF(IDIS(IA)) 200,150,200                                        RELABL44        1838
 150   IJ=IJ+1                                                         RELABL45        1839
       IDIS(IA)=L                                                      RELABL46        1840
       KO=KO+1                                                         RELABL47        1841
       IAJ(IJ)=IA                                                      RELABL48        1842
       IW(IJ)=IDEG(IA)                                                 RELABL49        1843
 200   CONTINUE                                                        RELABL50        1844
       IF(IJ-1)250,210,220                                             RELABL51        1845
 210   J=KO                                                            RELABL52        1846
       IZ=IAJ(1)                                                       RELABL53        1847
       NEW(KO)=IZ                                                      RELABL54        1848
       ILD(IZ)=KO                                                      RELABL55        1849
       GO TO 250                                                       RELABL56        1850
 220   X=0                                                             RELABL57        1851
 221   DO 230 I=2,IJ                                                   RELABL58        1852
       IF(IW(I)-IW(I-1))224,230,230                                    RELABL59        1853
 224   CONTINUE                                                        RELABL60        1854
       X=IW(I)                                                         RELABL61        1855
       IW(I)=IW(I-1)                                                   RELABL62        1856
       IW(I-1)=X                                                       RELABL63        1857
 225   X=IAJ(I)                                                        RELABL64        1858
       IAJ(I)=IAJ(I-1)                                                 RELABL65        1859
       IAJ(I-1)=X                                                      RELABL66        1860
 230   CONTINUE                                                        RELABL67        1861
       IF(X)235,235,220                                                RELABL68        1862
 235   DO 240 I=1,IJ                                                   RELABL69        1863
       J=J+1                                                           RELABL70        1864
       IZ=IAJ(I)                                                       RELABL71        1865
       NEW(J)=IZ                                                       RELABL72        1866
       ILD(IZ)=J                                                       RELABL73        1867
 240   CONTINUE                                                        RELABL74        1868
 250   IF(KO-NNC)130,255,255                                           RELABL75        1869
 255   CONTINUE                                                        RELABL76        1870
       RETURN                                                          RELABL77        1871
       END                                                             RELABL78        1872
       FUNCTION IDIST(NS,ML,MAXLEV,IG,II1,IC,IDEG,IDIS,IW,ICC)         IDIST  2        1873
C THIS FUNCTION HAS AS ITS VALUE THE MAXIMUM DISTANCE OF ANY NODE      IDIST  3        1874
C      IN COMPONENT IC(NS) FROM THE NODE NS.                           IDIST  4        1875
C THE DISTANCE OF EACH NODE IN THIS COMPONENT IS STORED IN THE ARRAY    IDIST  5        1876
C            IDIS.                                                     IDIST  6        1877
C THE MAXIMUM NUMBER OF NODES AT THE SAME DISTANCE FROM NS IS          IDIST  7        1878
C       STORED IN ML.                                                  IDIST  8        1879
       DIMENSION IG(II1,1),IC(1),IDEG(1),IDIS(1),IW(1),ICC(1)          IDIST  9        1880
       COMMON /S/ NN,MM,IH,IB                                          IDIST 10        1881
       COMMON /A/ MAXGRD                                               IDIST 11        1882
       COMMON /BITS/ NBITIN,NBITEX,IPASS                               IDIST 12        1883
       ICN=IC(NS)                                                      IDIST 13        1884
       NNC=ICC(ICN+1)-ICC(ICN)                                         IDIST 14        1885
       DO 50 I=1,NN                                                    IDIST 15        1886
       IF(IC(I)-IC(NS)) 50,40,50                                       IDIST 16        1887
 40    IDIS(I)=0                                                       IDIST 17        1888
   50  CONTINUE                                                        IDIST 18        1889
       LL=1                                                            IDIST 19        1890
       L=0                                                             IDIST 20        1891
       KI=0                                                            IDIST 21        1892
       KO=1                                                            IDIST 22        1893
       ML=0                                                            IDIST 23        1894
       IW(1)=NS                                                        IDIST 24        1895
       IDIS(NS)=-1                                                     IDIST 25        1896
  130  KI=KI+1                                                         IDIST 26        1897
       IF(KI-LL)135,132,135                                            IDIST 27        1898
  132  L=L+1                                                           IDIST 28        1899
       LL=KO+1                                                         IDIST 29        1900
```

```
      K=KO-KI+1                                                  IDIST 30    1901
      IF(K-ML) 135,135,133                                       IDIST 31    1902
  133 ML=K                                                       IDIST 32    1903
      IF(ML-MAXLEV) 135,135,220                                  IDIST 33    1904
135    II=IW(KI)                                                 IDIST 34    1905
      N=IDEG(II)                                                 IDIST 35    1906
      IF(N)140,215,140                                           IDIST 36    1907
  140 DO 200 I=1,N                                               IDIST 37    1908
      IA = IG(II,I)                                              JJ    47    1909
      IF(IDIS(IA))200,150,200                                    IDIST 39    1910
150   IDIS(IA)=L                                                 IDIST 40    1911
      KO=KO+1                                                    IDIST 41    1912
      IW(KO)=IA                                                  IDIST 42    1913
  200 CONTINUE                                                   IDIST 43    1914
      IF(KO-NNC)130,205,205                                      IDIST 44    1915
  205 IDIST=L                                                    IDIST 45    1916
      IDIS(NS)=0                                                 IDIST 46    1917
      K=KO-KI                                                    IDIST 47    1918
      IF(K-ML) 206,206,207                                       IDIST 48    1919
  207 ML=K                                                       IDIST 49    1920
  206 CONTINUE                                                   IDIST 50    1921
      RETURN                                                     IDIST 51    1922
  215 L=0                                                        IDIST 52    1923
      GO TO 205                                                  IDIST 53    1924
  220 IDIST=1                                                    IDIST 54    1925
      RETURN                                                     IDIST 55    1926
      END                                                        IDIST 56    1927
      SUBROUTINE REMARK(A)                                       JJ    48    1928
      RETURN                                                     JJ    49    1929
      END                                                        JJ    50    1930
      FUNCTION EOF(I)                                            JJ    51    1931
      INTEGER EOF                                                JJ    52    1932
      EOF=0                                                      JJ    53    1933
      RETURN                                                     JJ    54    1934
      END                                                        JJ    55    1935
```

INITIAL DISTRIBUTION

Copies

1 DNL

1 CNO

4 ONR
- 1 N. Basdekas
- 1 N. Perrone
- 1 S. Brodsky
- 1 R.J. Lundegard

1 ONR Chicago
- 1 R.N. Buchal

4 NRL
- 1 D. Curtis
- 1 R. Perlut
- 1 R.W. Alvarez
- 1 Lib

1 NAVMATCOMHQ
- 1 Mrs. S. Atchison

2 CDR, NAVORDSYSCOM
- 1 O. Seidman
- 1 Lib

3 NAVSHIPSYSCOMHQ
- 1 Dr. J.H. Huth
- 1 R.R. Kolesar
- 1 Tech Lib

2 CO, NAVAIRDEVCEN
- 1 A. Somoroff
- 1 Lib

2 CO, NAVCIVENGRLAB
- 1 J. Crawford
- 1 Lib

2 CDR, NELC
- 1 E. Nissan
- 1 Tech Lib

1 CDR, NAVOCEANO
- 1 Tech Lib

Copies

6 CDR, NOL
- 1 R.J. Edwards
- 1 W.A. Walker
- 1 J.F. Goeller
- 1 W. Wassmann
- 1 E.P. Johnson
- 1 Tech Lib

1 CO, NSSNF
- 1 Tech Lib

1 CO, NAVTRADEVCEN

2 NAVUSEARANDCEN HAWAII
- 1 A.T. Strickland
- 1 Lib

2 NAVUSEARANDCEN PASADENA
- 1 C.F. Falkenback
- 1 Lib

4 NAVUSEARANDCEN SAN DIEGO
- 1 J.T. Hunt
- 1 L. McCleary
- 1 D. Barach
- 1 Lib

1 NUSC NEWPT

4 NUSC NLON
- 1 A. Carlson
- 1 R. Manstan
- 1 J.W. Frye
- 1 R. Dunham

4 CDR, NAVWPNSCEN
- 1 J. Serpanos
- 1 D.E. Zilmer
- 1 W.J. Stronge
- 1 Tech Lib

4 CDR, NAVWPNSLAB
- 1 C.M. Blackmon
- 1 J. Schwartz
- 1 P. Johnson
- 1 Tech Lib

Copies	
3	SUPT, USNA 1 Dept of Math 1 Aerospace Dept 1 Tech Lib
1	SUPT, PGSCHOL, Monterey 1 Lib, Tech Reports Sec
1	ROTC & NAVADMINU MIT
1	Naval War College
5	CDR, NAVSEC 1 N. Griest 1 R.P. Lavoie 1 G.J. Snyder 1 T.W. Hull 1 Tech Lib
2	CDR, NAVAIRSYSCOMHQ 1 G.P. Maggos 1 Tech Lib
4	CDR, NAVFACENGCOMHQ 1 M. Yachnis 1 H.D. Nickerson 1 M.B. Hermann 1 Tech Lib
2	CO, NAVAIRENGCEN 1 T. Mazella 1 Tech Lib
1	CO, NAVAERORECOVFAC
1	CO, NAVAIRPROPTESTCEN
1	CDR, NAVAIRTESTCEN 1 Tech Lib
1	CO, NAVAIRTESTFAC
1	Naval Avionics Facility Indianapolis 1 S. Stephen
1	CO, NAVEODFAC
1	CO, NAVORDMISTESTFAC 1 Tech Lib

Copies	
1	CO, NAVORDTESTU
1	NAVSHIPYD BREM
1	NAVSHIPYD BSN
2	NAVSHIPYD CHASN 1 J.B. Kruse
1	NAVSHIPYD HUNTERS PT
1	NAVSHIPYD LBEACH
2	NAVSHIPYD MARE ISLAND 1 W. Richardson
1	NAVSHIPYD NORVA
1	NAVSHIPYD PEARL
1	NAVSHIPYD PHILA
2	NAVSHIPYD PTSMH 1 E.A. Fernald
1	CO, NAVSPASYSACT
1	NAVWPNSENGSUPTACT
1	CO, NAVWPNEVALFAC
1	CDR, PACMISTRAN
1	CDR, NAVMISCEN 1 Tech Lib
2	DIA, Washington, D.C. 20301 1 K. Goering 1 R. Wood
4	US ARMY, Harry Diamond Labs. 1 J. Miller 1 G.J. Hutchins 1 P.J. Emmerman 1 C. Anstine, Jr.
12	DDC

Copies

1 US Army Electronics Command, Fort Monmouth New Jersey 07703
  1 A. L. Sigismondi

5 USCOGARDHQ
  1 W. A. Cleary
  1 R. Johnson
  1 LCDR V. F. Keith
  1 LTjg M. C. Grosteck
  1 LT J. C. Card

5 USAMERDC
  1 H. G. Schaeffer
  1 J. Marburger
  1 R. W. Helmke
  1 C. L. Orth
  1 J. McDonald

5 NASA Greenbelt
  1 J. Mason
  1 W. Case
  1 T. G. Butler
  1 R. Courtney
  1 W. H. Browne, Jr.

4 NASA Langley Field
  1 J. P. Raney
  1 D. Weidman
  1 J. H. Starnes, Jr.
  1 O. U. Storaasli

1 NASA Ames
  1 P. R. Wilcox

1 NASA Huntsville
  1 S. M. Seltzer

5 ORL, Penn State
  1 J. C. Conway
  1 J. Kiusalass
  1 J. W. Sharer
  1 A. L. Stiehl
  1 Lib

Copies

1 JHU/APL
  1 W. Caywood
  1 G. Dailey
  1 R. M. Rivello
  1 J. S. O'Connor

1 University of Maryland
  1 Library

1 Mr. T. M. Olsen
Teledyne Isotopes
Energy Systems Division
110 W. Timonium Road
Lutherville, Maryland 21093

1 Mr. S. D. Hansen
The Boeing Company
Renton, Washington

1 Mr. Charles L. Blackburn
AVCO Aerostructures Div.
Nashville, Tennessee

1 Mr. Walter Tonelli
Convair Div. of General Dynamics, Kearny Villa Plant
Dept. 622-4
San Diego, California 92112

1 Dr. Robert H. Mallett
Advanced Reactor Div.
Westinghouse Electric Corp.
P. O. Box 158
Madison, Pennsylvania 15663

Copies

1 Dr. E.G. Baker
American Bureau of Shipping
45 Broad Street
New York, N.Y. 10004

1 Mr. R.B. Morgan
Texas Instruments, Inc.
P.O. Box 5012, MS 3
Dallas, Texas 75222

1 Mr. W.L. Cook
Room 6097
Communications Satellite Corp.
950 L'Enfant Plaza, S.W.
Washington, D.C. 20006

1 Mrs. P.L. Anderson
Advanced Analytical Tech Dept
Ford Motor Company
23400 Michigan Ave
Dearborn, Michigan 48124

1 Mr. C. Gerber
Dept. K678
McDonnell-Douglas Automation Co.
Box 516
St. Louis, Mo. 63166

1 Mr. R. Hopkins
Hercules, Inc.
Salt Lake City, Utah

1 Dr. Martin Goldberg
208 Violet Street
Massapequa Park, N.Y. 11762

1 Dr. R.H. MacNeal
MacNeal-Schwendler Corp.
7442 N. Figueroa Street
Los Angeles, Calif. 90041

1 Mr. W.S. Viall
Teledyne Brown Engr. Co., Inc.
Huntsville, Alabama

Copies

1 Dr. Richard Eppink
Civil Engineering Dept.
University of Virginia
Charlottesville, Va. 22901

1 Mr. Michael Hirtle
Aerospace Engineering Dept.
University of Maryland
College Park, Md. 20742

1 Mr. M. Pakstys
EB Div., Gen Dyn Corp.
Groton, Connecticut 06340

1 Dr. Roy Levy
Jet Propulsion Lab.
Calif. Inst. of Technology
Pasadena, Calif. 91103

1 Mr. P.R. Spencer (43-50)
Rm. 5130, Nassif Bldg.
National Highway Traffic
Safety Admin.
400-7th Street, S.W.
Washington, D.C. 20590

1 Mr. G.P. O'Hara
Applied Math. & Mech. Div.
Bldg. 124
Watervliet Arsenal
Watervliet, N.Y. 12189

1 Mr. R.K. Gieseke
Convair Aero. Div. of Gen Dyn
San Diego, California

1 Mr. Ronald P. Schmitz
Space Support Division
Sperry Rand Corp.
Huntsville, Alabama

CENTER DISTRIBUTION

Copies	Code
1	01
1	11 W.M. Ellsworth
1	15 Dr. W.E. Cummins
1	16 Dr. H.R. Chaplin
1	17 Dr. W.W. Murray
1	172
1	1721
1	1725
1	1727
1	173
1	1731
1	1735
1	174
1	1745
1	177 (UERD PTSMH)
1	177B (UERD PTSMH)
1	1775 (UERD PTSMH)
1	18 G. Gleissner
1	1805
1	184
1	1842
100	1844
1	185
1	1852
1	1853
1	1854
1	19 P.S. Dell Aria
1	194
1	196
1	1962
1	27 H.V. Nutt
1	271
1	272
1	273
1	2731
1	274
1	2741
1	2742
1	275
1	276
1	28 D.H. Kallas
1	287
1	94 J.L. Decker
1	9424 Dr. M.E. Lunchick
1	5641
1	5642
1	5643
1	5644

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)	2a. REPORT SECURITY CLASSIFICATION
Naval Ship R&D Center Bethesda, Maryland 20034	UNCLASSIFIED
	2b. GROUP

3. REPORT TITLE

The BANDIT Computer Program for the Reduction of Matrix Bandwidth for NASTRAN

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Research and Development Report

5. AUTHOR(S) (First name, middle initial, last name)

Gordon C. Everstine

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
March 1972	74	4

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO. ZR 0140201	Report 3827
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Naval Ship R&D Center Technical Director Bethesda, Maryland 20034

13. ABSTRACT

This report describes a **matrix** bandwidth reduction preprocessor for use with the NASA structural analysis computer program, NASTRAN. Called BANDIT, the program is written in FORTRAN and uses the Cuthill-McKee strategy for resequencing grid points. Versions of the program for both CDC and other computers are presented.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6801

UNCLASSIFIED

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
NASTRAN						
Finite Element Analysis						
Matrix Methods						
Bandwidth Reduction						
BANDIT						
Matrix Bandwidth Reduction						
Structural Analysis-Computer Methods						